

THESE présentée  
pour l'obtention  
du titre  
de DOCTEUR de l'UNIVERSITE  
de PARIS  
(mention Sciences)  
à  
l'UNIVERSITE de PARIS VI

Spécialité : MATHEMATIQUES

Option : INFORMATIQUE

par M. Abdelmajid BAKHTAOUI

Sujet de la thèse :

ALGORITHMES DE FORETS MINIMALES , RESEAUX DE TRANSPORT ET OPTIMALITE

Soutenue le 27 Juin 1972 devant la Commission composée de :

J.VILLE

Président

C.F.PICARD

Examineurs

R.PALLU DE LA BARRIERE



THESE présentée  
pour l'obtention  
du titre  
de DOCTEUR de l'UNIVERSITE  
de PARIS  
(mention Sciences)  
à  
l'UNIVERSITE de PARIS VI

Spécialité : MATHEMATIQUES

Option : INFORMATIQUE

par M. Abdelmajid BAKHTAOUI

Sujet de la thèse :

ALGORITHMES DE FORETS MINIMALES , RESEAUX DE TRANSPORT ET OPTIMALITE

Soutenue le 27 Juin 1972 devant la Commission composée de :

*J.VILLE*

Président

*C.F.PICARD*

Examineurs

*R.PALLU DE LA BARRIERE*



*Je remercie vivement Monsieur le Professeur VILLE de l'honneur qu'il me fait de présider le Jury de cette thèse.*

*J'exprime toute ma gratitude à Monsieur le Professeur PICARD qui a bien voulu diriger mes travaux de recherche et qui , par ses précieux conseils et encouragements m'a permis de mener à terme cette étude.*

*Je sais gré à Monsieur le Professeur PALLU DE LA BARRIERE d'avoir accepté de faire partie de ce Jury.*

*Je tiens à remercier Monsieur PUECHBERTY , Directeur Général de SOMEPOST , pour l'accueil chaleureux qu'il a bien voulu me réserver personnellement dès le premier jour.*

*J'exprime toute ma reconnaissance à Monsieur GALIMARD , Directeur de SOMEPOST , d'avoir eu en particulier l'obligeance de me faciliter , par l'intermédiaire du Chef de la Division INGENIERIE GENERALE , la rédaction de cette thèse.*

*Mes remerciements vont aussi à Madame THUILLIER pour le soin qu'elle a apporté à la présentation de cet ouvrage ainsi qu'à tous ceux qui, dans l'Equipe de Recherche STRUCTURES DE L'INFORMATION du C.N.R.S. ou dans les Equipes de SOMEPOST , en particulier celle des INGENIEURS-SYSTEME , m'ont aidé en m'apportant leurs critiques lors de nombreuses séances de travail.*



Digitized by the Internet Archive  
in 2026

[https://archive.org/details/IA41547103\\_0002](https://archive.org/details/IA41547103_0002)

## TABLE DES MATIERES

|                                                                       | Pages |
|-----------------------------------------------------------------------|-------|
| <u>PRESENTATION</u>                                                   | VI    |
| <u>CHAPITRE 1 - INTRODUCTION</u>                                      | 1     |
| 1.1. <u>FORMULATION DU PROBLEME FONDAMENTAL</u>                       | 3     |
| 1.1.1. Définition intuitive d'un graphe                               | 3     |
| 1.1.2. Enoncé du problème                                             | 3     |
| 1.1.3. Difficultés du problème                                        | 5     |
| 1.2. <u>PRINCIPE ET METHODE DE RESOLUTION DU PROBLEME</u>             | 5     |
| 1.2.1. Principe                                                       | 5     |
| 1.2.2. Méthode                                                        | 5     |
| <u>CHAPITRE 2 - TERMINOLOGIE ET PRESENTATION DES AUTRES CHAPITRES</u> | 7     |
| 2.1. <u>TERMINOLOGIE</u>                                              | 7     |
| 2.1.1. Quelques notions de la théorie des graphes                     | 7     |
| 2.1.2. Graphes particuliers                                           | 10    |
| 2.1.3. Termes propres à cet ouvrage                                   | 11    |
| 2.2. <u>PROPRIETES CARACTERISTIQUES DU GRAPHE G</u>                   | 13    |
| 2.3. <u>PRESENTATION DES AUTRES CHAPITRES</u>                         | 14    |
| 2.3.1. Arbres minimaux (chapitre 3)                                   | 14    |
| 2.3.2. Forêts minimales (chapitre 4)                                  | 14    |
| 2.3.3. Applications postales (chapitre 5)                             | 14    |
| 2.3.4. Programmation (chapitre 6)                                     | 14    |
| 2.3.5. Références                                                     | 14    |



|                                                                                          |        |
|------------------------------------------------------------------------------------------|--------|
| <u>CHAPITRE 3 - ARBRES MINIMAUX</u>                                                      | 15     |
| 3.1. <u>ARBRE MINIMAL D'OTOKAR BORUVKA</u>                                               | 17     |
| 3.2. <u>ARBRE MINIMAL DE J.B.KRUSKAL</u>                                                 | 17     |
| 3.2.1. Kruskal I                                                                         | 17     |
| 3.2.1.1 - Principes fondamentaux                                                         | 17     |
| 3.2.2. Kruskal II                                                                        | 18     |
| 3.2.3. Kruskal III                                                                       | 18     |
| 3.2.4. Kruskal III généralisée                                                           | 19     |
| 3.2.5. Kruskal IV                                                                        | 19     |
| 3.2.5. Remarque importante                                                               | 19     |
| 3.3. <u>ARBRE MINIMAL OBTENU PAR LA METHODE DE</u><br><u>"BOUNDED BRANCH AND BOUND "</u> | 20     |
| 3.3.1. Principe et propriétés de la méthode<br>(BBB)                                     | 21     |
| 3.3.2. Applications de la méthode                                                        | 21     |
| 3.3.2.1 - Commentaire de la figure 4<br>- Algorithme BBB -                               | 23     |
| 3.3.3. Remarques sur la méthode BBB                                                      | 23     |
| <br><u>CHAPITRE 4 - FORETS MINIMALES</u>                                                 | <br>25 |
| 4.A. <u>PROBLEME FONDAMENTAL ( [ 8 ] problème 3)</u>                                     | 27     |
| 4.A.1. Enoncé du problème                                                                | 27     |
| 4.A.1.1 - Définition de la minimalité<br>de cette forêt                                  | 28     |
| 4.A.2. Données et présentation détaillée du<br>problème                                  | 28     |
| 4.A.2.1 - Données                                                                        | 28     |
| 4.A.2.1 - Présentation du problème                                                       | 29     |
| 4.A.3. Résolution du problème à travers<br>différents exemples (fig. 6,7,8)              | 29     |
| 4.A.3.1 - Exemple 1                                                                      | 30     |
| 4.A.3.2 - Exemple 2                                                                      | 32     |
| 4.A.3.3 - Exemple 3                                                                      | 33     |



|                                                                                                           |    |
|-----------------------------------------------------------------------------------------------------------|----|
| 4.A.4. Formalisation et démonstration du théorème<br>fondamental                                          | 34 |
| 4.A.4.1 - Algorithme de forêt minimale                                                                    | 35 |
| 4.A.4.2 - Lemme 2                                                                                         | 36 |
| 4.A.4.3 - Théorème fondamental                                                                            | 36 |
| 4.A.4.3.1. Première variante :                                                                            |    |
| Cas des valuations toutes distinctes                                                                      | 36 |
| 4.A.4.3.2. Seconde variante :                                                                             |    |
| Cas des valuations pouvant être égales                                                                    | 39 |
| <br>4.B. <u>VARIANTES DU PROBLEME FONDAMENTAL</u>                                                         | 40 |
| 4.B.1. Problème 1 ( $C=\emptyset$ et $Z=X$ )                                                              | 41 |
| 4.B.2. Problème 1 bis ( $C=\emptyset$ et $Z=X$ )                                                          | 42 |
| 4.B.3. Problème 2 ( $\text{Card}(C_i)=1$ et $Z=X-C$ )                                                     | 42 |
| 4.B.4. Problème 2 bis ( $\text{Card}(C_i)=1 \forall i$ et $Z=X-C$ )                                       | 43 |
| 4.B.5. Problème 2 ter ( $\text{Card}(C_i)=1 \forall i$ et $Z=X-C$ )                                       | 44 |
| 4.B.6. Problème fondamental - Seconde variante                                                            | 46 |
| <br>4.C. <u>PROBLEMES POUVANT ADMETTRE LA PARTITION DES ENSEMBLES <math>C_i</math></u>                    | 46 |
| 4.C.1. Problème 5                                                                                         | 46 |
| 4.C.1.1 - $q = q' = p$                                                                                    | 47 |
| 4.C.1.2 - $q = q' < p$                                                                                    | 47 |
| 4.C.1.3 - $q < q' = p$                                                                                    | 48 |
| 4.C.1.4 - $q < q' < p$                                                                                    | 48 |
| 4.C.2. Problème 7                                                                                         | 48 |
| <br>4.D. <u>CONDITION SUFFISANTE POUR QUE F ADMETTE n-p ARCS ET<br/>SOIT SOLUTION DU PROBLEME 3 [ 8 ]</u> | 49 |
| <br>4.E. <u>CONCLUSION</u>                                                                                | 53 |
| <br><u>CHAPITRE 5 - APPLICATIONS POSTALES</u>                                                             | 55 |
| 5.1. <u>INTRODUCTION</u>                                                                                  | 55 |
| 5.2. <u>EXEMPLES D'APPLICATION DE L'ARBRE MINIMAL ET DE LA<br/>FORET MINIMALE</u>                         | 58 |



|                                                                                                                |    |
|----------------------------------------------------------------------------------------------------------------|----|
| 5.2.1. Problème A : Arbre minimal - Réseau d'acheminement du courrier de la Région Postale de Clermont-Ferrand | 58 |
| 5.2.2. Problème B : Meilleur rattachement d'un quantal postal                                                  | 60 |
| 5.2.2.1 - Procédure d'application                                                                              | 60 |
| 5.2.2.2 - Résultats                                                                                            | 61 |
| 5.2.3. Problème C : Forêt minimale - découpage du territoire en régions postales                               | 64 |
| 5.2.3.1 - Remarque 1                                                                                           | 67 |
| 5.2.3.2 - Remarque 2                                                                                           | 67 |
| 5.2.3.3 - Programmation                                                                                        | 67 |
| 5.3. <u>PROBLEME C BIS : FORET MINIMALE - PROCEDURE EXHAUSTIVE</u>                                             | 67 |
| 5.4. <u>PROBLEME D : ALIMENTATION FONCTIONNELLE D'UN PÔLE</u>                                                  | 69 |
| 5.4.1. Enoncé                                                                                                  | 69 |
| 5.4.2. Arborescence minimale de base variable - alimentation fonctionnelle d'un pôle                           | 69 |
| 5.4.3. Procédure d'application des arborescences                                                               | 72 |
| 5.5. <u>PROBLEME E : PROCEDURE POSTALE DE DETERMINATION DE LA "ZONE D'INFLUENCE"</u>                           | 75 |
| 5.5.1. Enoncé                                                                                                  | 75 |
| 5.5.2. Principe de résolution                                                                                  | 76 |
| 5.5.3. Remarques                                                                                               | 76 |
| 5.5.4. Exemple d'application                                                                                   | 76 |

## CHAPITRE 6 - PROGRAMMATION 77

|                                          |    |
|------------------------------------------|----|
| 6.1. <u>ALGORITHME DE FORET MINIMALE</u> | 77 |
| 6.1.1. Notation                          | 77 |
| 6.1.2. Programme de forêt minimale       | 78 |
| 6.1.3. Exemple numérique                 | 84 |



|                                                                |    |
|----------------------------------------------------------------|----|
| 6.2. <u>PROCEDURE EXHAUSTIVE DE VERIFICATION DES RESULTATS</u> | 85 |
| <u>DE LA FORET MINIMALE (§ 6.1.)</u>                           |    |
| 6.2.1. Principe                                                | 85 |
| 6.2.2. Remarques                                               | 85 |
| 6.2.3. Programme de la procédure exhaustive                    | 87 |
| 6.2.4. Exemple numérique                                       | 94 |
| <u>REFERENCES</u>                                              | 95 |



## P R E S E N T A T I O N

Tour à tour sous l'impulsion de plusieurs disciplines, essentiellement en recherche opérationnelle puis plus récemment en informatique théorique , la théorie des graphes a connu une rapide et prodigieuse évolution pendant cette dernière décennie.

Elle s'avère en effet être l'outil le mieux adapté à l'étude des configurations, c'est-à-dire la Combinatoire, soumises à différentes contraintes fixées à l'avance.

Elle nous a permis , grâce à des remarques bien appropriées , de résoudre certains problèmes d'organisation à partir de nouveaux concepts dont les résultats originaux ont été largement discutés et testés sur des problèmes réels (chapitres 5 et 6).

L'objectif de cet ouvrage est la résolution de problèmes de décomposition optimale (au sens des sommets) d'un graphe  $G$  et de minimisation de longueur associée à une forêt , graphe partiel du graphe  $G$  soumis à diverses contraintes de connexité.

Pour faciliter l'étude on était conduit à introduire une terminologie spécifique aux problèmes étudiés (§ 2.1.3.).

L'ouvrage se termine en proposant quelques voies de recherche.



## CHAPITRE I

### I N T R O D U C T I O N

Ce premier Chapitre, volontairement rédigé en termes intuitifs, est destiné aux non-mathématiciens de l'organisation et de la mécanisation postale, auprès de qui la théorie des graphes trouve un champ d'application tout récent et plein de promesses.

Cette présentation simplifiée<sup>(\*)</sup> leur permet de saisir rapidement la formulation et la résolution d'un des problèmes qui a pris naissance dans l'organisation postale, et d'entrevoir, avant même de poursuivre la lecture de la thèse, quelques unes des applications pratiques de son algorithme solution.

Ce problème concerne en outre de nombreux domaines tels que ceux de l'aéronautique, de l'irrigation, de la construction des routes, des réseaux de transport, de l'organisation des ateliers de production etc ...

Ce n'est qu'à partir du second chapitre, et sauf parfois le chapitre 5 "APPLICATIONS POSTALES", que les mathématiciens trouveront les termes auxquels ils sont habitués dans la théorie des graphes.

---

(\*) dont quelques idées sont reprises au chapitre 4 .



## 1.1. FORMULATION DU PROBLEME FONDAMENTAL.

### 1.1.1. Définition intuitive d'un graphe.

Ensemble de points et de flèches d'une figure géométrique.

Les points représentent les sommets<sup>(\*)</sup>, les flèches indiquent les liaisons, entre deux sommets, appelées arcs.

On notera  $n$  le nombre de sommets et  $m$  le nombre d'arcs du graphe. Les sommets sont indicés par un nombre compris entre zéro exclu et  $n$  fini ; les arcs en revanche sont valués par un nombre allant de zéro à l'infini.

### 1.1.2. Enoncé du problème.

Certains sommets fixés  $c_1, \dots, c_p$  appelés centres engendrent par une procédure non mathématique, de voisinage, trouvant son origine dans l'exploitation postale, une "zone d'influence"<sup>(\*\*)</sup> autour de chacun d'eux. Quelques sommets du graphe représentatif de toutes les "zones d'influence" se trouvent de la sorte connectés à plusieurs sommets fixés  $c_i$ . (Le cas où il existe des sommets non rattachables à aucun  $c_i$  est exclu).

#### Première partie de l'énoncé.

Pour fixer les idées prenons le graphe de la figure 1 représentatif de deux "zones d'influence", le raisonnement à  $p$  zones est strictement similaire ( $p > 2$ ).

La question qui se pose est de savoir déterminer la décomposition optimale<sup>(\*\*\*)</sup> du graphe en deux ensembles disjoints puisqu'il s'agit seulement de deux sommets fixés  $c_i$  (X5 et X21).

(\*) Bureaux de poste par exemple.

(\*\*) Se rapporter à la définition mathématique § 2.1.3.

(\*\*\*) Tout autre décomposition par rapport à cette décomposition optimale donnera une forêt minimale dont la somme des longueurs de tous ses arcs est de longueur plus grande.

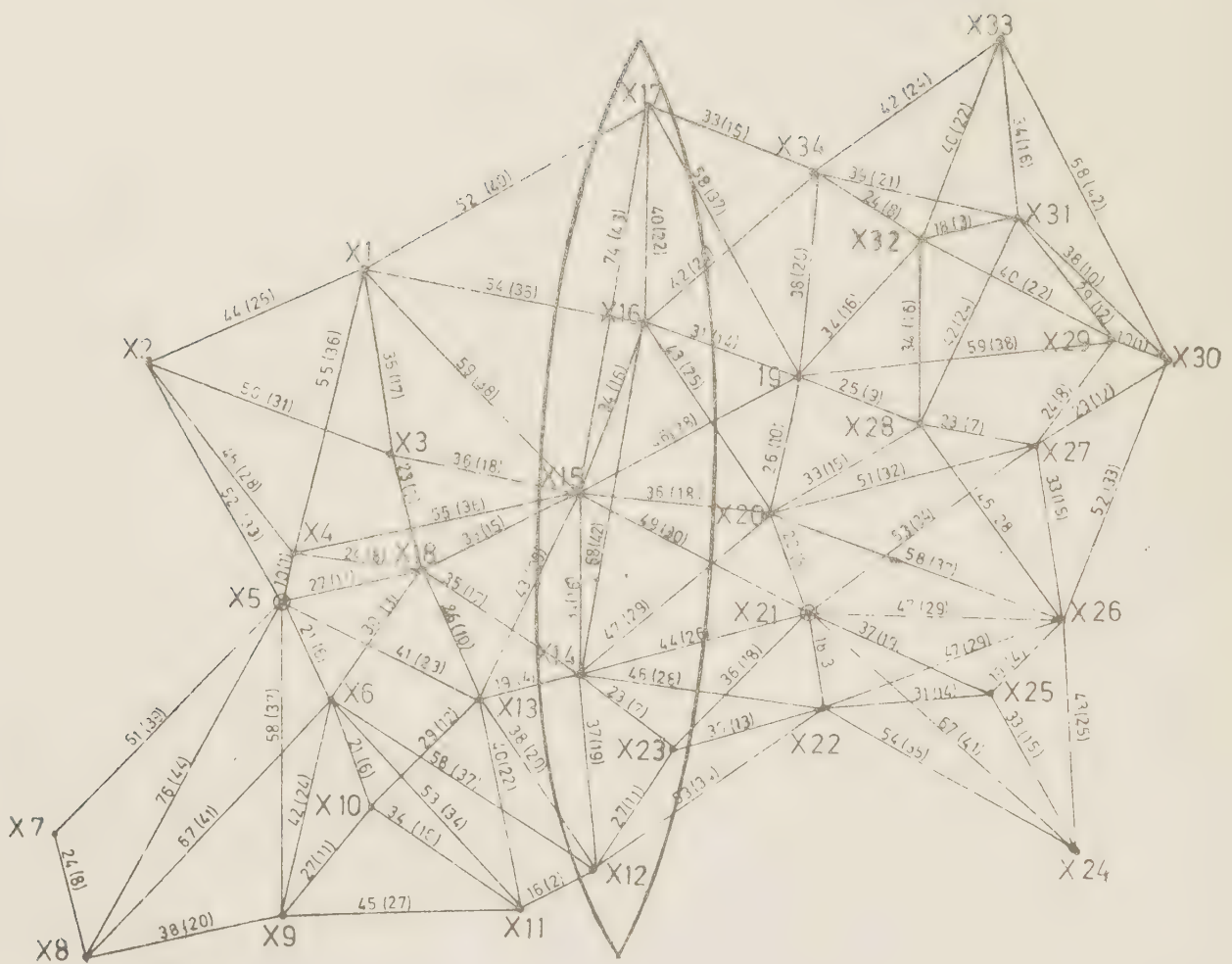


Figure 1 : Les sommets enfermés peuvent être rattachés à deux centres fixés  $c_i$  distincts : X5 et X21 .

Les liaisons jugées trop longues pour être sélectionnées par le calculateur ou liant deux sommets appartenant à deux "zones d'influence" distinctes, en dehors de leur intersection, ne sont pas représentées dans cette figure.

Les sommets rattachables simultanément à X5 et à X21 doivent être connectés à l'un de ces deux  $c_i$  .

D'après la figure 1 l'on peut à partir de chaque sommet fixé rejoindre par une chaîne tous les autres sommets du graphe.

### Seconde partie de l'énoncé.

Entre deux sommets du graphe il ne doit exister qu'un chemin et un seul (le plus court ou l'un des plus courts). L'ensemble de ces chemins ont une distance de longueur minimale, Le choix de ces chemins ne pose pas de problème, en revanche :

#### 1.1.3. Difficultés du problème.

- Il n'existe pas de procédure mathématique permettant de résoudre cette seconde partie du problème tant que l'on ne connaît pas , comme c'est le cas ici, tous les sommets rattachés exclusivement à un  $c_i$  donné ; autrement dit tant que l'on ne sait pas résoudre la première partie du problème.
- Si nous décomposons d'abord le graphe (1ère partie) puis nous recherchons les distances minimales, cela implique une minimalité du graphe partiel obtenu uniquement vis-à-vis de cette décomposition. Ce ne sera donc pas forcément le minimum absolu (\*) et c'est là un obstacle supplémentaire,

### 1.2. PRINCIPE ET METHODE DE RESOLUTION DU PROBLEME.

#### 1.2.1. Principe.

La procédure trouvée qui sera un exposé au chapitre 3 (§3.A. 4.1.) résoudra simultanément les deux parties du problème.

#### 1.2.2. Méthode.

Soit une "zone de recouvrement" constituée par l'ensemble des sommets  $X_{12}$ ,  $X_{14}$ ,  $X_{15}$ ,  $X_{16}$ ,  $X_{17}$  et  $X_{23}$  de la figure 1 qui sont rattachés simultanément aux deux sommets fixés  $c_i$  ( $X_5$  et  $X_{21}$ ).

---

(\*) aux contraintes du problème près.

Les difficultés du problème consistent :

- à savoir à quel Centre  $c_i$  il faut rattacher la "zone de recouvrement" .
- à savoir s'il faut la rattacher en partie ou en totalité à un  $c_i$  donné.
- à connaître la procédure de sélection des arcs de chaque arbre<sup>(\*)</sup>, procédure qui semble impossible puisqu'on ne connaît pas au préalable tous les sommets qui le composent.

On trouvera au chapitre 4 les réponses rigoureuses à ces questions.

---

(\*) Arbre = graphe connexe et sans cycle . Un graphe est dit connexe si entre deux quelconques de ses sommets il existe toujours une chaîne (un trait continu formé par une succession d'arcs).

## CHAPITRE II

### TERMINOLOGIE ET PRESENTATION DES AUTRES CHAPITRES

#### 2.1. TERMINOLOGIE

Tous les termes de graphe employés dans cette thèse sont compris au sens de PICARD [ 1 ] .

Il s'agira toujours de graphes finis.

Soit  $G = (X, U, P_U)$  où :

$X$  est l'ensemble des sommets du graphe  $G$  .

$U$  est l'ensemble de ses arcs ,

$P_U : U \rightarrow \mathbb{R}^+$  est la fonction valuation des arcs du graphe.

#### 2.1.1. Quelques notions de base de la théorie des graphes.

Sous-graphe  $S$  de  $G$  .

Soit  $G = (X, U)$  et  $S = (X', U')$  avec  $X' \in X$  et

$U' = \{z \mid z \in U \text{ et } pr_1 z \in X' \text{ et } pr_2 z \in X'\}$  (\*)

c'est-à-dire  $U' = U \cap X'^2$

Graphe partiel de  $G$  .

graphe  $P = (X, U'')$  avec  $U'' \subset U$

(\*)  $pr_1 z$  = première projection de  $z$

$pr_2 z$  = deuxième " de  $z$  .

### Sous-graphe partiel de G .

C'est  $P = (X', U'')$  de  $G = (X, U)$  où

$$X' \subset X \quad \text{et} \quad U'' \subset U'$$

c'est-à-dire un graphe partiel  $(X', U'')$  d'un sous-graphe  $(X', U')$ .

### Couple :

La relation binaire  $\exists x, \exists y$  telle que  $z = (x, y)$  s'appelle un couple où :

$$x = \text{pr}_1 z \quad \text{et} \quad y = \text{pr}_2 z .$$

### Boucle :

Si  $\text{pr}_1 z = \text{pr}_2 z$  alors l'arc  $z$  est une boucle.

### Chemin.

Ensemble d'arcs orientés dans le même sens et ordonnés : l'extrémité initiale de chaque arc - sauf du premier - est extrémité terminale du précédent.

### Chemin non simple.

Famille d'arcs, distincts ou non, ordonnés comme dans un chemin.

### Chemin simple.

Tous les arcs sont distincts dans le chemin.

### Chemin élémentaire.

Chemin dont les origines des arcs sont distincts.

### Circuit.

Chemin dont l'extrémité est confondue avec l'origine.

### Chaîne.

Ensemble d'arcs dont l'extrémité initiale de l'un est extrémité initiale ou terminale du précédent ou du suivant, tout sommet sauf

les extrêmes appartient à deux couples consécutifs.

### Cycle.

Chaîne dont l'extrémité est confondue avec l'origine.

### Cycle élémentaire.

Cycle ne passant pas deux fois par le même sommet.

### Degré $\delta_i$ d'un sommet $i$ .

Nombre d'arcs qui ont ce sommet  $i$  pour extrémité initiale ou terminale; une boucle étant comptée deux fois.

### Demi-degré extérieur $\delta_i^+$ .

Appelé aussi base, c'est le nombre d'arcs sortant d'un sommet  $i$ .

### Demi-degré intérieur $\delta_i^-$ .

Nombre d'arcs entrant dans un sommet  $i$ .

### Descendant d'un sommet $a$ .

Sommet extrémité d'un chemin ayant le sommet  $a$  pour origine.

### Ascendant d'un sommet $b$ .

Sommet origine d'un chemin ayant le sommet  $b$  pour extrémité.

### Cocycle $\omega(A)$ .

Ensemble d'arcs ayant une extrémité et une seule dans  $A$ .

$$(\omega(A) = \omega^+(A) + \omega^-(A) \text{ où :}$$

$$\omega^+(A) = \text{constitué par les arcs sortant de } A$$

$$\omega^-(A) = \text{ " " " " entrant dans } A).$$

### Circuit $\omega^+(A)$ .

Ensemble des arcs sortant d'un sous-ensemble  $A$  de sommets d'un graphe  $(X, U)$  et entrant dans  $X-A$ , lorsqu'aucun arc n'entre dans  $A$ .

Cocycle élémentaire.

Formé des arcs de liaison entre deux sous-graphes connexes l'un et l'autre.

Cocircuit élémentaire.

Lorsque tous les arcs de liaison, entre deux sous-graphes connexes l'un et l'autre, entrent dans le même sous-graphe de sommets considérés.

Nombre cyclomatique  $K(G)$ .

$$l'entier \quad K(G) = m - n + p$$

$m$  = nombre d'arcs de  $G$

$n$  = nombre de sommets de  $G$

$p$  = nombre de composantes connexes de  $G$ .

Nombre cocyclomatique  $\ell(G)$ .

$$\ell(G) = n - p$$

Fermeture transitive du graphe.

$\forall x \in X$  dans le graphe  $(X, U)$   $\hat{f}_x$  comprend tous les descendants de  $x$ .

2.1.2. Graphes particuliers.Graphe complet.

Il existe au moins un arc liant deux sommets quelconque de ce graphe :

$$\forall x, \forall y (x \neq y) : (x, y) \notin U \Rightarrow (y, x) \in U.$$

Graphe fini.

Si l'ensemble  $X$  des sommets du graphe  $G$  est fini :  $|X| = n$  s'appelle ordre du graphe quand  $G$  est fini.

Graphe connexe.

ou encore faiblement connexe. Entre deux sommets quelconques de ce graphe il existe toujours une chaîne.

Graphe q.f.c.i.

S'il existe un sommet ascendant de tous les autres, alors le graphe est quasi-fortement connexe inférieurement (q.f.c.i.)

Graphe symétrique.

La relation  $(x,y) \in U \Rightarrow (y,x) \in U$  est toujours vraie.

Composante connexe.

Sous-graphe connexe isolé du reste du graphe.

Arbre.

Graphe connexe et sans cycles.

Coarbre  $K = (X,W)$  de  $G = (X,U)$  connexe.

Tout graphe partiel de  $G$  qui ne contient pas de cocycle élémentaire de  $G$  et tel que tout arc  $u \in U-W$  introduise un cocycle élémentaire dans  $G$ .

Arborescence.

Graphe q.f.c.i. et sans cycle.

Racine.

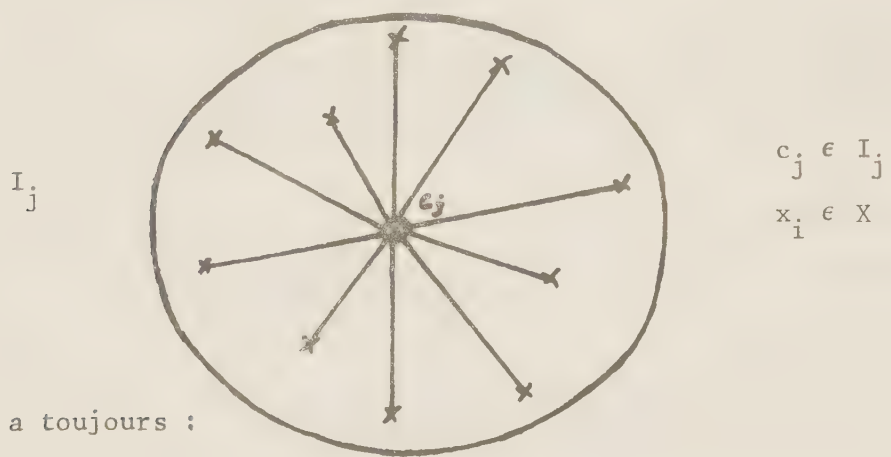
Tout sommet sans ascendant véritable.

2.1.3. Termes propres à cet ouvrage.

La nécessité d'exprimer quelques idées spécifiques aux problèmes posés nous a conduit toutefois à donner quelques définitions suivantes et à souligner quelques propriétés caractéristiques du graphe  $G$ .

• "Zone d'Influence"  $I_j$  :

Soit une famille indicée  $(I_j)_{j \in N^*}$  d'ensembles de sommets où  $I_j$  appelé "zone d'influence" est représenté comme suit :



on a toujours :

- a)  $|I_j| \geq 2$
- b) pour  $j$  fixé, il existe un et un seul sommet  $c_j$ , appelé centre, qui est lié à tout autre élément de  $I_j$  par un arc.

On notera que  $\bigcup_{j=1}^{\ell} I_j = X$  ( $\ell$  entier naturel fini) et que les  $I_j$  peuvent être considérés comme des fermés comprenant chacun un nombre fini de sommets, qui recouvrent le graphe  $G = (X, U, P_U)$ .

• "Zone de recouvrement"  $Z$  :

Si  $I_j \cap I_i \neq \emptyset$  (\*) alors cette intersection sera notée  $Z_s$  ;  
soit  $Z = \bigcup_{s=1}^r Z_s$ ,  $r \leq \ell \cdot \frac{\ell-1}{2}$ .  $Z$  s'appelle "zone de recouvrement". Elle est constituée d'éléments de  $X$  qui peuvent être connectés par un arc simultanément à au moins deux centres  $c_i, c_j$  distincts.

• "Zone propre"  $C_i$  :

Sous-ensemble de sommets de la "zone d'influence" correspondante  $I_i$  tel que :

---

(\*) on suppose que leur intersection ne contient un  $c_i$ , ni  $c_j \quad \forall i \forall j$

$$I_i = C_i \cup Z_i$$

et

$$C_j \cap Z_i = \emptyset$$

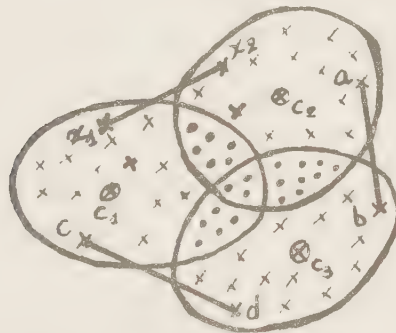
avec  $Z_i \subset Z$  et  $C_i \subset C$ ,  $C = \bigcup_{i=1}^p C_i$ .

On notera aussi :

$$Z \cup C = X \quad \text{et} \quad Z \cap C = \emptyset.$$

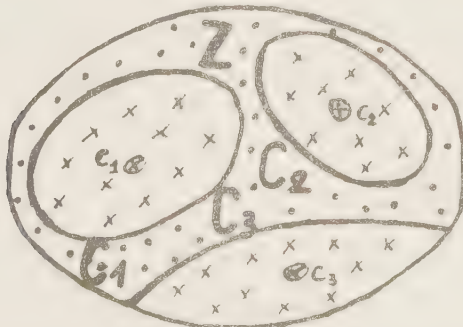
## 2.2. PROPRIETES CARACTERISTIQUES DU GRAPHE $G$ .

- (1)  $C_i \cap C_j = \emptyset \quad \forall i \neq j$
- (2)  $I_i \cap I_j \neq \emptyset$  (sauf spécification contraire).
- (3) Si le graphe donné  $G_0^{(*)}$  ne vérifie pas (1) alors il faudra lui substituer un graphe  $G$  par suppression de tout arc liant des sommets appartenant à deux "zones propres"  $C_i, C_j$  distinctes.



Graphe  $G_0$

Après suppression des arcs  $(x_1, x_2)$ ,  $(a, b)$  et  $(c, d)$ , on obtient le graphe  $G$  qu'on peut schématiser plus simplement :



Graphe  $G$

On adoptera dorénavant cette représentation.

(\*) pour la clarté du schéma, on n'a pas représenté les arcs des graphes  $G_0$  et  $G$ .

## 2.3. PRESENTATION DES AUTRES CHAPITRES.

### 2.3.1. Arbres minimaux (chapitre 3) :

On y rencontrera un bref rappel des travaux de J.B.Kruskal présentés par P.Rosenstiehl ainsi que la construction de l'arbre minimal par la méthode de Branch and bound.

### 2.3.2. Forêts minimales (chapitre 4) :

Le premier paragraphe traite du problème fondamental, le second présente d'autres problèmes de forêts minimales, variantes du précédent; quant au paragraphe C , on y **abordera** des problèmes qui admettent cette fois la partition des ensembles  $C_i$  .

### 2.3.3. Applications Postales (chapitre 5) :

Survol très bref de quelques exemples d'applications de la théorie des graphes dans l'organisation de la collecte et de la distribution du courrier ainsi que dans l'alimentation optimale d'un centre de traitement du courrier.

### 2.3.4. Programmation (chapitre 6) :

On y trouvera le programme de l'algorithme de forêt minimale et celui d'une procédure exhaustive qui aboutit aux mêmes résultats.

Cette deuxième procédure sert de moyen de vérification de l'optimalité de la forêt minimale à titre d'illustration de la démonstration.

### 2.3.5. Références.

## CHAPITRE III

### ARBRES MINIMAUX

Ce chapitre fait un bref rappel de quelques unes des procédures de la construction de l'arbre minimal établies depuis les travaux d'Otokar BORUVKA [ 3 ] .

Ces procédures, ainsi que celles connues à ce jour, ne permettent pas de résoudre le problème que nous avons dégagé d'une réalité industrielle<sup>(\*)</sup>.

Cela a donné lieu à une voie de recherche qui a débouché sur la généralisation de la construction A de KRUSKAL appelée - KRUSKAL I - par P.ROSENSTIEHL (cf. chapitre 4).

Au cours de ces travaux, il a été constaté que la méthode "Boundary Branch and Bound peut également servir à construire l'arbre minimal. Un bref exposé de la méthode et un exemple seront donnés dans ce chapitre.

---

(\*) Optimalité dans le découpage du territoire et dans l'organisation de l'ensemble des réseaux d'acheminement du courrier.



### 3.1. ARBRE MINIMAL D'OTOKAR BORUVKA.

Avant Kruskal, Boruvka [ 3 ] a établi en se servant des matrices le théorème suivant :

"Si un graphe connexe, ayant un nombre fini de points est valué (valuation des arcs) ; si ses arcs sont tous de longueurs distinctes , alors parmi tous les arbres issus du graphe, il existe un arbre et un seul dont la somme des longueurs des arcs est minimale".

### 3.2. ARBRE MINIMAL DE J.B. KRUSKAL.

Kruskal [ 2 ] a repris les travaux d'O.Boruvka en donnant trois constructions simples de l'arbre minimal et en soupçonnant l'existence d'une quatrième que P.Rosenstiehl [ 4 ] a présenté en 1966.

On passera brièvement en revue ces quatre constructions en partant du graphe  $G = (X, U, P_U)$  valué - valuation des arcs - .

#### 3.2.1. KRUSKAL I (\*)

Données : - la liste des arcs de  $G$  ordonnés dans l'ordre croissant de leur longueur,

- pour chaque arc ses deux extrémités.

Principe : Tout arc lu ne constituant pas de cycle avec les arcs retenus est retenu.

Les  $n-1$  arcs (\*\*) sélectionnés constituent l'arbre minimal.

#### 3.2.1.1. Principes Fondamentaux.

Cet algorithme et ceux qui suivent dans ce chapitre sont tous

(\*) P.Rosenstiehl appelle Kruskal I, II et III respectivement les constructions A , A' et B de Kruskal.

(\*\*) le graphe  $G$  est connexe. Ce ne sera pas le cas dans Kruskal II.

soumis aux deux conditions suivantes [ 4 ] :

- Une propriété caractéristique des arcs de l'arbre minimal d'un graphe  $G$  est d'être premier arc dans au moins un cocycle de  $G$ .
- Une propriété caractéristique des arcs de l'arbre minimal d'un graphe  $G$  est de ne jamais être dernier arc dans les cycles de  $G$ .

### 3.2.2. KRUSKAL II .

Cette méthode est duale de Kruskal I .

Données : - La liste des arcs de  $G$  ordonnés dans l'ordre décroissant,  
- et pour chaque arc ses deux extrémités.

Principe : Tout arc lu est supprimé si le nombre de composantes du graphe  $G$  reste constant. Les arcs lus et non supprimés constituent l'arbre minimal de  $n-p$  arcs ( $p \geq 1$ ).

### 3.2.3. KRUSKAL III.

Données : Les cocycles  $w(A)$  pour  $A = \{x\} \forall x \in X$ .

Principe : On constitue une liste partielle de cocycles donnés. Cette liste sera modifiée sans augmenter en longueur jusqu'à devenir vide. Au début dans  $w(A)$ ,  $A = \{x\}$ , par suite  $A$  est une partie quelconque de  $X$ .

On retient le premier arc  $u$  d'un cocycle  $w(A)$  de la liste partielle. Une seule extrémité de  $u \in A$ ; l'autre  $t$  peut appartenir à un ensemble  $B$  pour lequel  $w(B)$  est de la liste partielle, et alors dans cette liste  $w(A) + w(B)$  vient remplacer  $w(A)$  et  $w(B)$ ; sinon on pose  $B = \{t\}$ , et dans la liste partielle  $w(A) + w(B)$  remplace  $w(A)$  uniquement.

Les arcs retenus quand la liste partielle s'avère vide, constituant l'arc minimal.

Afin d'introduire Kruskal IV, P. Rosenstiehl a algébrisé l'algorithme Kruskal III.

#### 3.2.4. KRUSKAL III GENERALISEE.

Données : une base  $B$  du groupe des cocycles de  $G^{(*)}$ .

Exemple : les cocycles associés à un arbre quelconque  $V$ .

Principe : A partir de  $B$ , en  $r$  étapes, construire une base minimale  $B_*$  comme indiqué au § III de [4] ; en déduire  $V_*$  somme des premiers éléments des cocycles de  $B_*$  où  $V_*$  est l'arbre cherché.

#### 3.2.5. KRUSKAL IV.

C'est P.ROSENSTIEHL qui a construit cet algorithme en substituant "cycle" à "cocycle" dans Kruskal III généralisé.

Il est parti de la judicieuse remarque que les procédures de calcul vont souvent par paire à la manière de "cycles et de cocycles".

Données : une base  $B'$  du groupe des cycles du graphe  $G$ .

Exemple : les cycles associés à un coarbre quelconque  $W$ .

Principe: A partir de  $B'$ , en  $s$  étapes, construire la base maximale  $B'^*$  comme indiqué au § III de [4] ; en déduire  $W^*$  et poser  $V_* = U - W^*$ , où  $W^*$  est le coarbre maximal de l'ensemble des coarbres de  $G$ .

#### 3.2.6. Remarque importante.

Dix huit ans avant la publication des travaux de J.B. KRUSKAL [2],

---

(\*) Se rapporter au § III de [4], pour la définition de  $B$ ,  $B'$ ,  $B_*$ ,  $B'^*$ ,  $V_*$  et  $W^*$  qu'on rencontrera dans Kruskal III généralisé et Kruskal IV.

le mathématicien G.CHOQUET [ 5 ] avait donné un algorithme de construction de l'arbre minimal semblable à celui de Kruskal III et qui servait à tracer le réseau de routes minimal entre  $n$  points situés dans un espace muni d'une métrique quelconque. Un peu plus tard C.R.PRIM [ 6 ] a repris les travaux de Kruskal en donnant sa propre version de l'algorithme ~~Kruskal~~ Kruskal III.

Dans le prochain paragraphe, on notera que la méthode de BRANCH AND BOUND permet également de construire l'arbre minimal de manière équivalente à celle de l'algorithme Kruskal III .

### 3.3. ARBRE MINIMAL OBTENU PAR LA METHODE DE "BOUNDED BRANCH AND BOUND" .

Comme il a été remarqué au § 3.2.6, il ne s'agit pas d'une nouvelle méthode de construction de l'arbre minimal mais de l'importante constatation suivante :

"La méthode de Branch and Bound , utilisée d'ordinaire dans la recherche de procédures heuristiques, permet également la construction de l'arbre *minimal*".

#### 3.3.1. Principe et Propriétés de la méthode (BBB) (\*)

C'est une parente de la méthode que Land et Doig [ 7 ] ont proposé en 1960 pour construire le chemin de valeur minimale joignant la racine  $S_0$  d'un graphe  $G$  au sommet terminal.

Propriétés de la Méthode BBB :

- permet de conserver en mémoire un nombre de sommets de l'arborescence relativement très faible et prévisible (\*\*);
- s'applique à une fonction économique et à des contraintes non linéaires ;

---

(\*) (BBB) est la notation par ABADIE [ 7 ] de la méthode "Bounded Branch and Bound".

(\*\*) On ne connaît pas a priori, ce qui est un défaut important pour la méthode de Land et Doig, une borne raisonnable du nombre de sommets de l'arborescence qu'il est nécessaire de conserver en mémoire du calculateur pendant les étapes intermédiaires.

- fournit rapidement, comme toutes les procédures heuristiques, une première solution discrète "acceptable".

### 3.3.2. Applications de la méthode.

Soit le graphe  $G = (X, U, P_U)$  connexe, fini et valué - valuation des arcs - ;

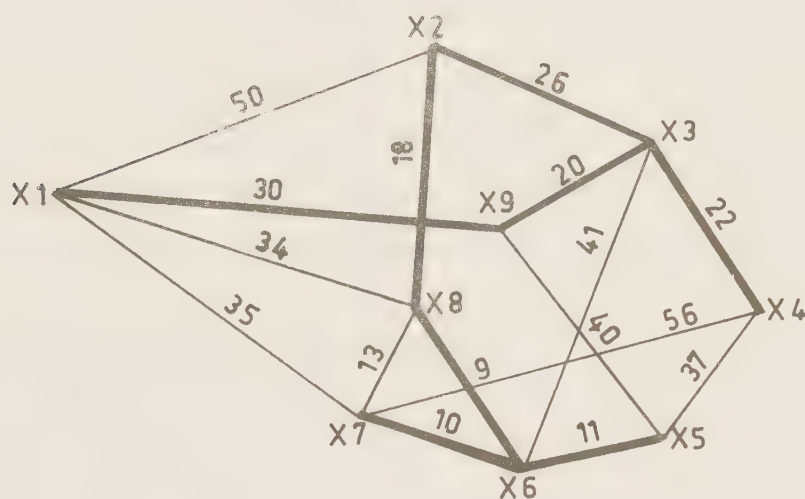


Figure 3 - Graphe connexe , fini et valué.

Les traits renforcés représentent les arcs de l'arbre minimal.

Le choix des arcs se fait selon le processus indiqué par la figure 4 suivante:

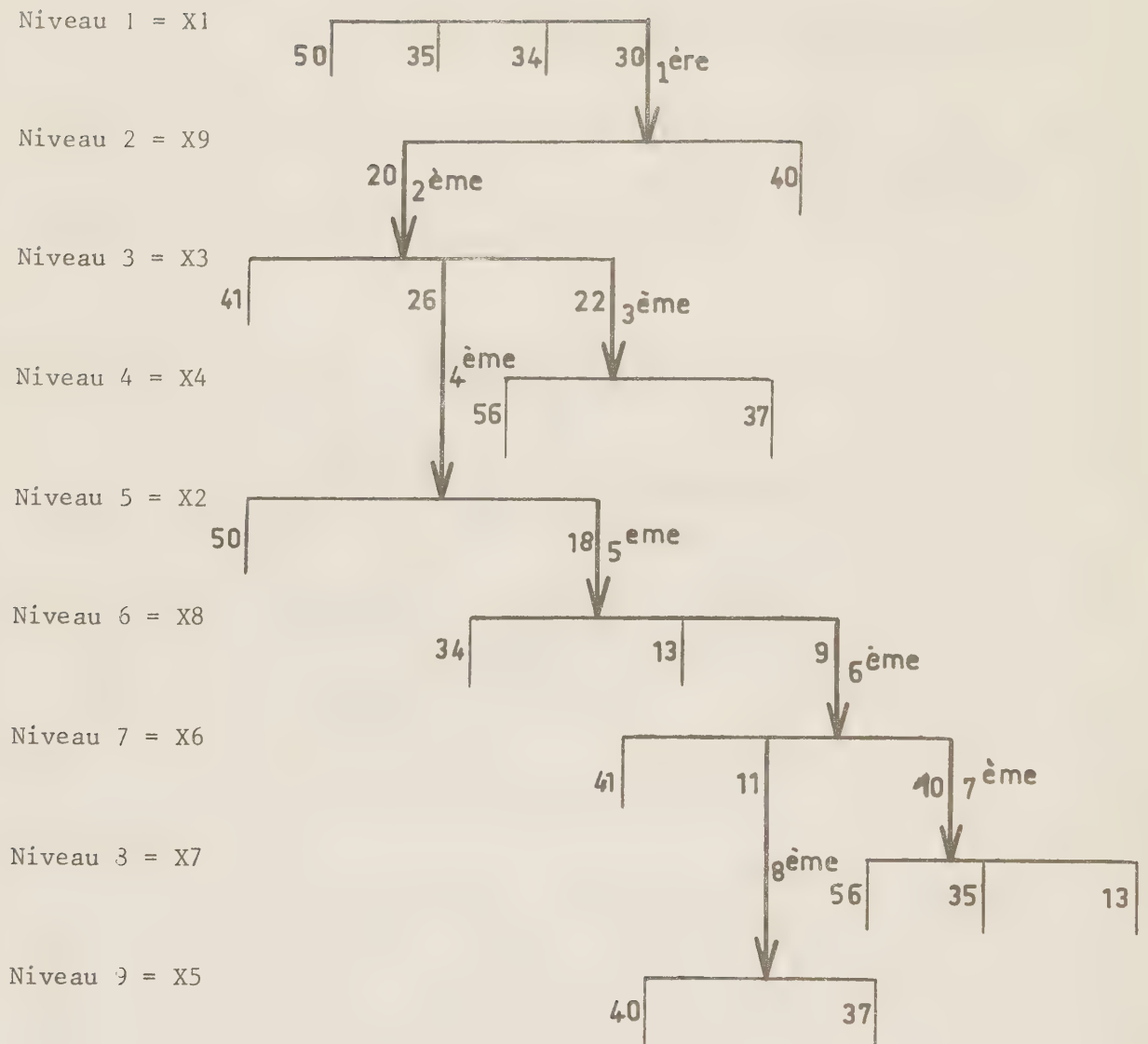


Figure 4 - Schéma de principe de la Méthode  
Branch and Bound

### 3.3.2.1 - Commentaire de la figure 4 - Algorithme BBB -

- Choisir en partant d'un sommet quelconque  $X_i \forall i$  avec  $i = 1, 2, \dots, 9$  la solution la meilleure<sup>(\*)</sup> qui conduira au sommet suivant  $X_j$  ;
- La sélection en  $X_j$  , niveau 2 , se fait parmi les arcs rejetés en  $X_i$  et tous ceux de  $X_j$  . Elle consiste à prendre l'arc le plus court;
- Si l'arc le plus court n'est pas issu de  $X_j$  (dernier sommet de la liste des sommets à partir desquels la sélection des arcs a déjà été faite) tous les successeurs de  $X_j$  ne sont pas retenus (niveaux 4 et 7);
- Poursuivre le processus jusqu'à ce que les arcs sélectionnés forment un graphe partiel connexe ou bien quand on a retenu  $n-1$  arcs;
- Le graphe partiel ainsi obtenu sera l'arbre minimal cherché.

### 3.3.3. Remarques sur la méthode BBB .

- 1) C'est une méthode de construction directe de l'arbre minimal : c'est-à-dire tout arc retenu ne fera l'objet d'aucun rejet au cours d'une phase ultérieure dans le déroulement du programme sur machine.
- 2) La méthode BBB ne se contente pas de fournir une solution discrète approchée comme toutes les procédures heuristiques, mais donne un résultat exact en permettant de construire l'arbre minimal recherché.
- 3) On notera pour conclure que la méthode BBB s'applique mieux à certains types de problèmes<sup>(\*\*)</sup> que celle de Kruskal I qui a par contre l'avantage d'être plus facilement généralisable pour construire une forêt minimale (chapitre 4).

(\*) Il s'agit de l'arc de plus courte longueur.

(\*\*) Par exemple celui où l'on donne la base d'un sommet (cf. 5.4.

"Applications postales")



## CHAPITRE IV

### FORETS MINIMALES

Les problèmes sont résolus par ordre chronologique dans ce chapitre.

La première partie (§A) est réservée au problème fondamental faisant l'objet de la thèse, quant à la seconde partie (§B), elle présente différents problèmes de recherches de forêts graphes partiels minimaux ; on résoudra d'abord les problèmes sans contraintes, puis ceux avec contraintes de sommets, voire même de contraintes d'ensembles de sommets  $C_1, \dots, C_q$ . Ces problèmes classés par ordre de difficulté croissante présentent toutes les conditions de rejet des arcs qu'on retrouve dans le problème fondamental. On verra enfin (§C) le cas des problèmes où l'on admet la partition des ensembles  $C_i$ .

Pour faciliter l'exposé de ce chapitre on est amené à souligner, dans cette introduction, les conventions ci-après de façon à ne pas les répéter tout le long du texte. Ces conventions sont d'ailleurs valables, dans tout l'ouvrage, sauf spécifications contraires :

- Le graphe  $G = (X, U, P_U)$  est fini d'ordre  $n$ , sans boucles, connexe ou non (\*) et valué par l'application  $P : U \rightarrow \overline{\mathbb{R}}^+$ .

Soit  $p_0 \geq 1$  le nombre de composantes connexes. Quand  $p_0 > 1$  il faut que  $p_0 \leq p$  où  $p$  est le nombre d'arbres de la forêt minimale recherchée. On prendra ici, pour simplification sans perte de généralité  $p_0 = 1$ .

- A chaque arbre on associe :
  - un sommet fixé  $c_i$  et un seul ;

---

(\*) On raisonnera toujours sur un graphe connexe, voire même complet, sans diminuer en rien la généralité des problèmes. Les algorithmes donnés seront également valables pour les graphes non connexes.

- un ensemble de sommets  $C_i$  et un seul (\*)

- L'ensemble  $C_i$  ne peut être partitionné. Il doit se retrouver intégralement dans un arbre de  $F$ .

Pour l'algorithme donnant la forêt  $F$  graphe partiel de  $G$ , le sommet fixé  $c_i$  joue le même rôle que l'un quelconque des sommets de  $C_i$ , "zone propre" engendrée par  $c_i$ .

- On suppose qu'on a un ordinateur pour extraire la forêt recherchée. Les données sont donc lues sur fichier. Les valuations des arcs sont en général ordonnées de façon non décroissante pour tenir compte du cas des arcs égaux.
- Chaque sommet fixé  $c_i$  engendre une "zone d'influence" (§2.1.) et une seule. L'intersection de ces zones, supposée toujours non vide (§5.2.C. Problème C), s'appelle "zone de recouvrement", cependant l'une des propriétés de l'algorithme de la forêt minimale (4.A.4.1.) permet de considérer l'ensemble de ces zones comme étant unique  $Z$  (\*\*). Quand on parlera de "zone de recouvrement", il s'agira donc toujours de l'ensemble de toutes les intersections.
- Le graphe  $G$  déduit du graphe  $G_0$  initial (§2.2) ne doit pas contenir d'arc reliant deux sommets appartenant à deux "zones d'influence" distinctes, sauf quand ces sommets se trouvent dans leur intersection. On passera ainsi en revue toutes les "zones d'influence" prises deux à deux afin d'éliminer ces arcs avant la constitution des données.

---

(\*) Chaque ensemble  $C_i$ , quand il existe, contient un sommet fixé  $c_i$  et un seul, par une hypothèse qui trouve sa justification dans des problèmes concrets. Chaque arbre contiendra en fin de compte  $C_i \supset c_i$ , et peut-être certains sommets de  $Z$  "zone de recouvrement".

(\*\*) Un sommet  $x_2$ , très éloigné du sommet  $x_1$  où l'on se trouve, appartenant à une autre intersection, peut être considéré de la même intersection puisque l'arc  $(x_1, x_2)$  supposé alors de longueur infinie ne sera jamais sélectionné.

#### 4.A. PROBLEME FONDAMENTAL. ([ 8 ] problème 3 ) (\*)

##### 4.A.1. Enoncé du Problème.

Soit le graphe  $G = (X, U, P_U)$  ayant  $p$  "zones propres"  $C_i$  avec  $p \geq 2$ .

Construire une forêt minimale de  $p$  composantes connexes telle que :

- des ensembles de sommets  $C_1, \dots, C_p$  tous non vides appartiennent à  $p$  composantes distinctes;
- des sommets  $z_1, \dots, z_r$  soient connectés à l'un des ensembles  $C_1, \dots, C_p$  à l'exclusion de tous les autres;
- les ensembles  $C_1, \dots, C_p$  appartiennent chacun à une des  $p$  composantes (Fig. 5).

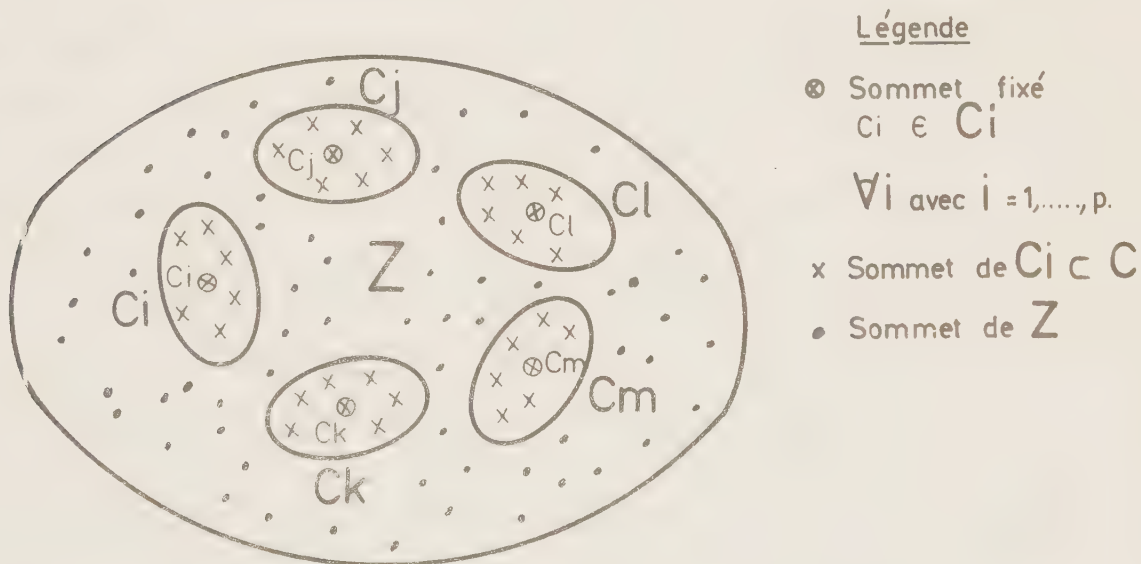


Figure 5 - Graphe G (représenté sans arc pour la clarté de la figure) (\*\*)

(\*) Les problèmes conservent les mêmes numéros que dans [ 8 ].

(\*\*) Il en sera de même pour les graphes qui suivent dans ce chapitre 4.

#### 4.A.1.1 - Définition de la minimalité de cette forêt.

A partir du graphe  $G = (X, U, P_U)$  de la Figure 5, il faut construire un ensemble d'arbres formant une forêt graphe partiel de  $G$ , dont la somme des longueurs des arcs soit minimale.

#### 4.A.2. Données et présentation détaillée du problème.

##### 4.A.2.1 - Données.

Les ensembles  $C_i$  de sommets ne sont pas initialement des données du problème, mais résultent de la procédure mentionnée au §1.1.2. qui trouve son origine dans un problème concret d'organisation (§ 5.5).

Les  $C_i$  sont de la sorte des ensembles de sommets, qui dans la forêt cherchée, sont obligatoirement connectés aux sommets correspondants  $c_i$ ; c'est-à-dire qu'à chaque  $c_i$  fixé on rattache un ensemble  $C_i$  et un seul.

On précise qu'aucun sommet de  $C_i$  n'appartient à la zone de recouvrement "Z" dont tous les points sont, rappelons le, susceptibles d'être connectés à au moins deux  $C_i$ , donc à au moins deux  $c_i$ , comme on le verra plus loin.

Les données deviennent donc à partir de  $G = (X, U, P_U)$  où :

- $X$  est l'ensemble des sommets du graphe  $G$  tel que
  - $C = \bigcup_{i=1}^p C_i$  = ensemble des "zones propres"  $C_i$  avec
 
$$\text{card}(C_i) \geq 1 \text{ et } C_i = \bigcup_{t=1}^l \{y_t\} \quad \forall t \in N^* \text{ fini ;}$$
  - $Z = \bigcup_{s=1}^r Z_s$  = ensemble des sommets de la "zone de recouvrement"
 
$$\text{avec } Z_s = \{z_s\} ;$$
  - $$\begin{cases} C \cup Z = X \\ C \cap Z = \emptyset . \end{cases}$$

- $U$  est l'ensemble des arcs du graphe  $G$  tel que
 
$$U = \bigcup_{j=1}^m u_j \quad \forall m \text{ fini};$$
- $P_U : U \rightarrow \overline{\mathbb{R}^+}$  est une fonction de valuation qui à chaque arc  $u_j \in U$  fait correspondre un nombre réel appartenant à l'intervalle  $]0, +\infty]$ .

#### 4.A.2.2 - Présentation du Problème.

Soit le graphe  $G = (X, U, P_U)$ . Le problème consiste à le partitionner en  $p$  composantes connexes et à en extraire une forêt selon la définition donnée au paragraphe 4.A.1.1.

Il s'agit d'une partition optimale du graphe  $G$  car toute autre partition imposée au graphe initial  $G$  permettra d'en extraire une forêt minimale uniquement vis-à-vis de cette partition.

Ainsi le problème de l'arbre minimal, résolu par plusieurs auteurs [2], [3], [4], [5], [6] ..., part d'un graphe  $G$  de  $p_0$  composantes ( $p_0 \geq 1$ ) et nécessite au préalable la donnée des nombres des sommets de chaque composante. La forêt obtenue de  $p$  arbres minimaux par l'une des procédures de ces auteurs, quand  $p_0 > 1$ , ne peut représenter la longueur minimale absolue extraite du graphe  $G$ , puisque la décomposition de  $G$  en  $p_0$  composantes est une donnée du problème et est tout à fait arbitraire (Problème 2 bis §4.B.3).

B.ROY [9] a d'ailleurs souligné dans un problème voisin de celui-là l'importance et la difficulté d'une meilleure décomposition du graphe  $G$ . Pour généraliser le problème de l'arbre minimal en l'étendant à une forêt on a dû faire appel (§4.A.1) à un certain nombre de contraintes qui permettent d'aboutir à la partition optimale du graphe  $G$ .

#### 4.A.3. Résolution du problème à travers différents exemples. (Fig.6,7,8)

La méthode utilisée pour construire la forêt est celle de

construction directe , c'est-à-dire tous les arcs retenus ici ne font l'objet d'aucun rejet au cours d'une phase ultérieure. Ces arcs sont ceux de la forêt minimale recherchée et qui est définie comme étant un graphe fini sans cycle dans lequel toute composante est un arbre ou un arbre dégénéré quand il ne contient qu'un sommet isolé.

On suppose que les arcs sont classés par ordre non décroissant  $(u_1) \leq l(u_2) \leq \dots \leq l(u_m)$ .

Dans la méthode utilisée à la note [8] sur les "Algorithmes de Forêts minimales" nous avons procédé par élimination des arcs à partir du graphe  $G$  donné. Dans celle-là on adoptera plutôt le processus de construction directe exposé ci-dessus.

En partant du graphe  $G$  , tout arc lu dans le fichier et répondant aux contraintes, est donc retenu; la suppression d'un arc n'intervient qu'au moment de son traitement dans le calculateur<sup>(\*)</sup>.

Soit  $(x_1, x_2)$  un arc à sélectionner ( $x_1$  et  $x_2$  distincts) de plus courte longueur parmi ceux non encore insérés dans la forêt. On verra de façon détaillée les exemples de sélection des arcs  $(C, C)$  ,  $(Z, Z)$  et  $(C, Z)$  de façon à mieux expliciter la définition du cocycle interçit qu'on rencontrera au paragraphe 4.A.4. Les arcs  $(C, C)$  ,  $(Z, Z)$  et  $(C, Z)$  ont leurs origines respectivement dans  $C, Z$  puis  $C$  et leurs extrémités respectivement dans  $C, Z$  et  $Z$  .

4.A.3.1. Exemple 1 : Sélection d'un arc  $(x_1, x_2)$  où  $x_1$  et  $x_2$  appartiennent à  $C$  .

- Si  $\exists i$   $x_1 \in C_i$  et  $x_2 \in C_i$  retenir l'arc  $(x_1, x_2)$  s'il ne crée pas de cycle. Ce sera le cas s'il n'existe pas de chaîne entre  $x_1$  et  $x_2$  .
- Si  $x_1$  (resp.  $x_2$ )  $\in C_i$  et  $x_2$  (resp.  $x_1$ )  $\in C_j$  ,  $j \neq i$  rejeter l'arc  $(x_1, x_2)$  . (Figure 6).

|                    |                                                                                                                                                                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <u>PROPRIÉTÉ 1</u> | Un graphe $G = (X, U, P_U)$ de $p \geq 2$ "zones propres" $C_i$ admet un graphe partiel qui soit une forêt de $p$ composantes connexes s'il n'existe pas d'arc, ou de chaînes, entre les sommets de deux $C_i$ distincts dans la forêt. |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

(\*) Notons toutefois que lorsqu'on sélectionne  $n-p$  arcs, il n'est pas nécessaire de poursuivre le traitement des arcs restant puisqu'ils seront rejetés d'office du fait que la valuation de chacun d'eux est plus grande (ou égale en créant un cycle).

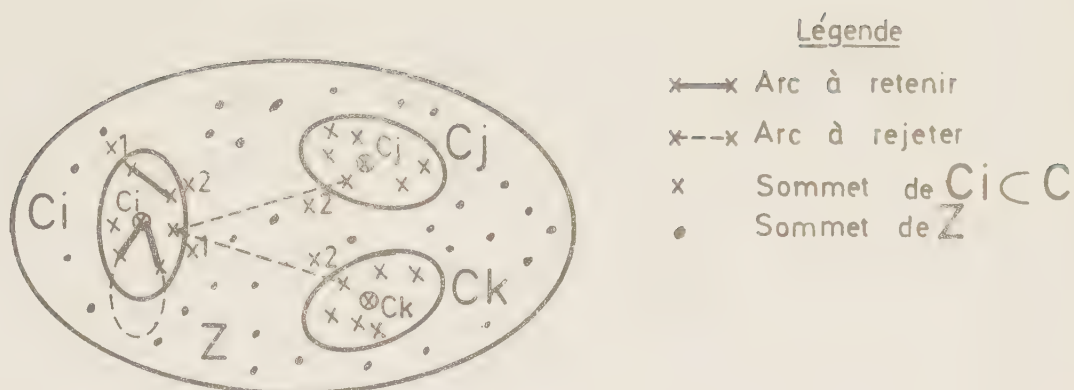


Figure 6 - Sélection d'un arc (C,C)

Démonstration :

Soit un graphe  $G$  de  $p$  composantes  $C_i$ . On cherche à construire une forêt de  $p$  arbres.

Supposons qu'il existe une chaîne entre un sommet de  $C_i$  et un autre de  $C_j$  ( $C_i \neq C_j$ ), comme chaque  $C_i$  engendre un arbre, la forêt obtenue aura donc  $p-1$  arbres ( $C_i$  et  $C_j$  engendrent le même arbre) ce qui est contradictoire avec l'hypothèse.

|                    |                                                                                                                                                                 |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <u>PROPRIÉTÉ 2</u> | Soit un graphe $G = (X, U, P_U)$ , <u>le nombre des "zones propres" <math>C_i</math> doit être toujours <math>\geq 2</math></u> pour construire une forêt $F$ . |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|

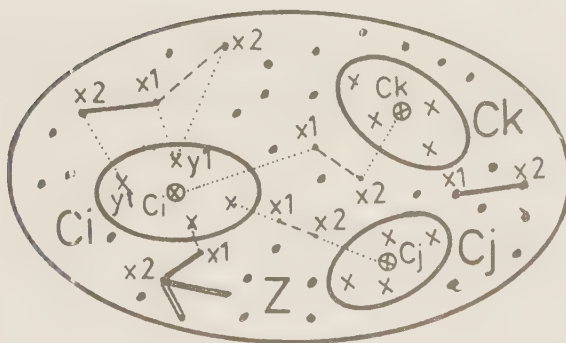
Rappelons qu'il s'agit de la forêt obéissant aux contraintes du §4.A.1. On sait que chaque arbre de  $F$  contient tous les sommets d'un même  $C_i$ . Si le graphe  $G$  admet zéro ou un  $C_i$ , il se pose le problème de l'existence de l'ensemble  $Z$ , "zone de recouvrement" dans laquelle tous les sommets sont connectables par définition à deux où plusieurs  $C_i$ .

Le graphe  $G$  a donc nécessairement un nombre de  $C_i \geq 2$ .

4.A.3.2. Exemple 2 - Sélection d'un arc  $(x_1, x_2)$  où  $x_1$  et  $x_2$  appartient à  $Z$ .

• Soient  $x_1$  et  $x_2 \in Z$

- Si  $y_t$  déjà sélectionné dans la forêt  $F$ , si  $x_1 \hat{r}_s y_t, y_t \in C_i$  et  $x_2 \hat{r}_s y_t, y_t \in C_j$  avec  $C_i \neq C_j$  alors rejeter l'arc  $(x_1, x_2)$ ;
- Si  $x_1 \hat{r}_s y_s, y_s \in C_i$  et  $x_2 \hat{r}_s y_t, y_t \in C_i$  :
  - + si  $y_s = y_t$  rejeter l'arc  $(x_1, x_2)$  qui crée ainsi un cycle.
  - + si  $y_s \neq y_t$  tel qu'il n'existe pas de chaîne entre  $y_s$  et  $y_t$  sélectionner l'arc  $(x_1, x_2)$  ;
- Si  $x_1 \hat{r}_s y_t, y_t \in C_i$  et  $y_t \in F$  retenir l'arc  $(x_1, x_2)$  et les  $x_i$  tels que  $(\forall i) x_i \hat{r}_s x_2$  si aucun des  $x_i$  n'est connecté à  $C_j \neq C_i$ .
- Si  $x_1$  et  $x_2 \in Z$  tel qu'il n'existe pas de chaîne entre les sommets de  $x_1, x_2$  et l'ensemble  $C_i \forall i$  alors retenir l'arc  $(x_1, x_2)$ .



Légende

- $x \cdots \cdots$  Arc déjà retenu
- $\cdots \cdots$  Arc à rejeter
- $\text{---}$  Arc à retenir
- $\text{---}$  Arc rattaché à un sommet donné de  $Z$
- $\bullet$  Sommet de  $Z$
- $x$  Sommet de  $C_i \subset C$

Figure 7 - Sélection d'un arc  $(Z, Z)$

Lemme 1 : Soit  $G = (X, U, P_U)$  un graphe connexe de  $p \geq 2$  "zones propres"  $C_i$ .

Entre deux  $C_i$  distincts, il n'existe aucune chaîne dans la forêt minimale  $F$ .

Supposons qu'il existe  $q$  chaînes ( $q \geq 1$ ) avec  $q < p$ , la forêt  $F$  sera formée de  $p-q$  arbres dont certains contiennent plus d'un  $C_i$ , ce qui est contraire aux hypothèses du problème (§ 4.A.1).

C.Q.F.D.

PROPRIÉTÉ 3

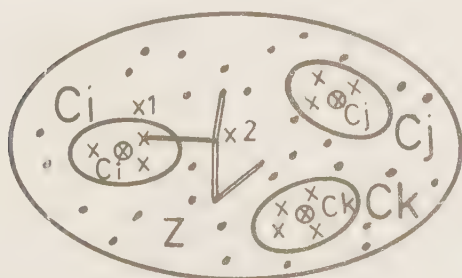
Soit  $G = (X, U, P_U)$  un graphe de  $p \geq 2$  "zones propres".

Tout sommet  $z_s \in Z$ , et tous ceux de sa classe faible  $Ez_s^{(*)}$  dans la forêt  $F$  peuvent être rattachés à un autre sommet de  $Z$  lié à une "zone propre"  $C_i$  si aucun des sommets de  $Ez_s$  n'est connecté à un  $C_j \neq C_i$ .

En vertu du lemme 1, il ne peut en être autrement, c'est-à-dire qu'on ne peut lier un élément de  $Z$  à deux  $C_i$  distincts, sinon on crée une chaîne entre ces deux  $C_i$  (contraire aux hypothèses).

4.A.3.3. Exemple 3 - Sélection d'un arc  $(x_1, x_2)$  où  $x_1 \in C$  et  $x_2 \in Z$ .

Soit  $(x_1, x_2)$  un arc à sélectionner. Si  $(i \neq j)x_1 \in C_i \subset C$  et  $x_2 \in Z$  et si entre le sommet  $x_2$  et l'ensemble  $C_j$  il n'existe aucune chaîne alors il faut retenir l'arc  $(x_1, x_2)$  et tous les sommets formant une chaîne avec  $x_2 \in Z$ ; c'est-à-dire appartenant à la même classe faible  $E_{x_2}$ .



Légende

- Arc à retenir
- Arc rattaché à un sommet donné de  $Z$
- $x$  Sommet de  $C_i \subset C$
- $\bullet$  Sommet de  $Z$

Figure 8 - Sélection d'un arc  $(C, Z)$

(\*) cf (1) définition 28, page 9.  $Ez_s$  est la classe faible de  $z_s$  (par l'équivalence de la connexité faible), représentant un ensemble de sommets.

PROPRIETE 4 | Soit  $G = (X, U, P_U)$  un graphe de  $p \geq 2$  "zones propres"  $C_i$ .  
 Tout sommet  $z_s$  de  $Z$ , et tous ceux de sa classe faible  $Ez_s$  dans la forêt  $F$ , peuvent être rattachés à une "zone propre" et à une seule.

Démonstration analogue à celle de la propriété 3.

Cette propriété 4 est plus forte que la propriété 3.

#### 4.A.4. Formalisation et démonstration du théorème fondamental.

Toute forêt  $F$  graphe partiel minimal de  $G = (X, U, P_U)$  telle qu'on la définit doit répondre, au cours de sa construction, aux contraintes des quatre propriétés précédentes et du lemme 1. Ces contraintes peuvent se résumer en utilisant la notion de cocycle interdit.

DEFINITION. On appelle cocycle interdit tout cocycle élémentaire qui enfreint une des quatre propriétés précitées.

PROPRIETE 5 | "Il ne doit pas exister de *cocycle interdit* entre les sommets de deux arbres distincts au cours de leur construction progressive dans la forêt  $F$ ".

On rappelle qu'il existe autant d'arbres  $A_i$ , dans la forêt  $F$ , que de "zones propres"  $C_i$  et que tous les sommets de chaque  $C_i$  appartiennent à un arbre  $A_i$  et à un seul. La notion de cocycle interdit peut s'exprimer autrement :

"Il ne doit pas exister de cocycle élémentaire entre  $A_i$  et  $A_j$   $\forall i \forall j$  ( $i \neq j$ ). Le symbole  $A_i$  représente deux notions différentes pour traduire :

- 1)  $A_i$  = un des arbres de la forêt minimale  $F$ ,
- 2)  $A_i$  = un sous-graphe partiel de l'arbre cité en 1).

On rappelle (chapitre III) que tout arc créant un cycle doit être rejeté lors de la construction de l'arbre minimal (condition  $\alpha$ ).

Avec les deux restrictions :

- pas de cocycle interdit ,
- pas de cycle ,

on peut poser l'algorithme direct donnant la forêt minimale  $F$  recherchée :

#### 4.A.4.1. Algorithme de forêt minimale.

Données : - la liste des arcs du graphe  $G = (X, U, P_U)$  ordonnée dans l'ordre non décroissant et pour tout arc ses deux extrémités;

- le nombre de "zones propres"  $C_i$ , quand elles existent, doit être plus grand ou égal à deux pour permettre l'existence de la "zone de recouvrement"  $Z^{(*)}$

Procédure : On lit la liste des arcs du graphe. Tout arc lu est sélectionné :

- s'il ne crée pas de cycle avec ceux déjà retenus dans la forêt  $F$  ,

et

- s'il ne crée pas de cocycle interdit entre deux sommets de deux arbres distincts  $A_i$  et  $A_j \forall i \forall j (i \neq j)$  au cours de leur construction progressive suivant l'ordre d'apparition des arcs.

Répéter la procédure autant que possible jusqu'à sélection de  $n-p$  arcs, c'est-à-dire connexité de chaque arbre  $A_i$  qui contient nécessairement<sup>(\*\*)</sup> tous les sommets d'un  $C_i$  et éventuellement des sommets de la "zone de recouvrement"  $Z$  .

Cet algorithme servira à la démonstration du théorème fondamental si l'on pose et démontre le lemme suivant :

(\*) § 4.B.1. Problème 1 où les  $C_i$  n'existent pas. (Dans le problème du § 4.A.1. les  $C_i$  existent évidemment).

(\*\*) l'algorithme reste valable avec ou sans l'hypothèse que "chaque arbre  $A_i$  contient tous les sommets d'un  $C_i$ " (§ 4C). Il doit remplir en outre les conditions suffisantes de (§ 4D)

4.A.4.2. Lemme 2.

Soit  $F = (X, A, P_A)$  un graphe fini d'ordre  $n$ , les propriétés suivantes sont équivalentes et caractérisent une forêt  $F$  de  $p$  arbres.

- (1)  $F$  est sans cycles et admet  $n-p$  arcs ;
- (2)  $F$  est sans cycles et en ajoutant un arc entre un couple de sommets on crée :
  - soit un cycle et un seul sur les sommets d'un arbre quelconque de  $F$  ,
  - soit un cocycle interdit et un seul entre les sommets de deux arbres distincts de  $F$  .

(1)  $\Rightarrow$  (2) Par hypothèse  $F$  est sans cycles ( $k(F) = m-n+p = 0$ ) et admet  $n-p$  arcs ( $l(F) = n-p$  arcs),

Si l'on ajoute un arc à  $F$  :

- ou bien le nombre cyclomatique  $k(F)$  sera égal à 1 , ce qui donne un cycle et un seul sur les sommets d'un arbre quelconque de  $F$  ,
- ou bien le nombre cocyclomatique  $l(F)$  augmente de 1, ce qui crée un cocycle interdit et un seul entre les sommets de deux arbres distincts de  $F$  .

(2)  $\Rightarrow$  (1) Par hypothèse  $F$  est sans cycles, c'est-à-dire  $k(F) = m-n+p = 0 \Rightarrow m = n-p$  , donc  $F$  admet  $n-p$  arcs.

4.A.4.3. Théorème fondamental.

4.A.4.3.1. Première variante : Cas des valuations toutes distinctes.

Soit  $G = (X, U, P_U)$  un graphe ayant  $p \geq 2$  "zones propres"  $C_i$  et une "zone de recouvrement" unique  $Z$  (\*)

Si les longueurs des arcs du graphe sont toutes de valeurs

---

(\*) rappelons que  $X = Z \cup C$  et  $Z \cap C = \emptyset$  où  $C = \bigcup_{i=1}^p C_i$

distinctes alors les p arbres obtenus par l'algorithme de forêt minimale sont uniques et constituent la forêt  $F = (X, A, P_A)$ , graphe partiel de  $G = (X, U, P_U)$ , dont la somme des valuations de tous ses arcs est minimale par rapport à toutes celles des autres forêts issues de  $G$ .

Il est évident que  $F = (X, A, P_A)$  sans cycles, -  $k(F) = m - n + p = 0$  -, et sans cocycles interdits, admet  $m = n - p$  arcs avec :

- $n$  = nombre des sommets du graphe  $G$ ,
- $p$  = nombre de composantes connexes de  $F$ ,

si la condition suffisante (§ 4.D) est remplie.

#### Démonstration

1)  $F = (X, A, P_A)$  est une forêt de  $n - p$  arcs.

Soient  $A_k = \{a_1, a_2, \dots, a_k\}$  les arcs acceptés dans l'ordre selon la procédure de l'algorithme de forêt minimale.

Supposons qu'à  $A_k$  on ne peut plus ajouter d'arc sans créer, de cycles sur les sommets d'un arbre quelconque, ou de cocycles interdits entre deux sommets de deux arbres distincts.

D'après le lemme 2 le graphe partiel  $F = (X, A, P_A)$  est une forêt (propriété 2) et admet  $k$  arcs (propriété 1) avec  $k = n - p'$ .

Si  $p' = p$  alors la forêt est minimale et elle est unique. Si  $p' \neq p$  ( $p < p'$ ) on a une forêt de  $p'$  composantes, il manque  $p' - p$  arcs pour résoudre le problème (blocage). Le problème a toutefois une solution s'il remplit la condition suffisante (§ 4.D.).

Il restera à démontrer que  $A$  (donc  $F$ ) est unique. C'est-à-dire que  $A = A_{n-p} = \{a_1, \dots, a_{n-p}\}$  et que la forêt est minimale.

2) Unicité de  $A$ .

De l'hypothèse que les arcs du graphe  $G$  ont tous des longueurs distinctes, il résulte que l'algorithme de forêt minimale procède de manière unique ; les  $\{a_i\}$  sont donc uniques, d'où l'unicité de  $A$ .

### 3) Minimalité de $F$ ( $A = A_{n-p}$ )

Raisonnons par l'absurde en partant d'une forêt  $F = (X, A, P_A)$  pour laquelle

$$\sum_{a_i \in A} l(a_i)^{(*)} \text{ soit minimale.}$$

Si  $A \neq A_{n-p} = \{a_1, a_2, \dots, a_{n-p}\}$ ,

soit  $a_k$  le premier arc tel que  $a_k \in A_{n-p}$  et  $a_k \notin A$ .

On déduit que  $A \supset \{a_1, a_2, \dots, a_{k-1}\} = A_{k-1}$ .

La réunion  $A \cup \{a_k\}$  est telle que (propriété 2)

- soit cette réunion crée un cycle  $\supset a_k$
- soit cette réunion crée un cocycle interdit  $\supset a_k$

3.a/  $A \cup \{a_k\}$  crée un cycle et un seul (propriété 2).

Ce cycle contient obligatoirement un arc  $e \notin A_{n-p}$  mais cet arc  $e \in A$ .

Posons :

$$W = (A \cup \{a_k\}) - \{e\}$$

Le graphe  $(X, W, P_W)$  est sans cycles et admet  $n-p$  arcs (puisque'en ajoutant un arc qui ne crée pas de cocycle interdit, on engendre un cycle et un seul).  $(X, W, P_W)$  est donc une forêt (propriété 1).

$A_{k-1} \cup \{e\}$  est sans cycles (car  $A_{k-1} \subset A$  et  $\{e\} \subset A$  soit  $A_{k-1} \cup \{e\} \subset A$  sans cycles par hypothèse puisque  $F = (X, A, P_A)$  est une forêt).

Selon le processus de l'algorithme utilisé, l'arc  $(e)$  intervient après l'arc  $(a_k)$  donc

(\*)  $l(a_i)$  = longueur de l'arc  $a_i$ .

$$\ell(e) > \ell(a_k)$$

La forêt  $(X, W, P_W)$  qui se déduit  $(X, A, P_A)$  par le seul changement de l'arc  $(e)$  en l'arc  $(a_k)$  a donc une longueur totale inférieure à celle de  $(X, A, P_A)$ . Cette forêt est telle que

$$(X, W, P_W) < (X, A, P_A)$$

mais  $(X, A, P_A)$  est la forêt minimale par hypothèse, ce qui est donc contradictoire et par suite

$$\underline{A = A_{n-p}} ,$$

3.b/  $A \cup \{a_k\}$  crée un cocycle interdit et un seul (propriété 2).

Le raisonnement est analogue à celui du paragraphe 3.a/ .

Il suffit de remplacer l'expression :

"ce cycle contient obligatoirement un arc  $e \notin A_{n-p}$ "

par l'expression :

"ce cocycle interdit contient obligatoirement un arc  $e \notin A_{n-p}$ "

cela implique que  $A = A_{n-p}$

On vient de voir que  $F = (X, A, P_A)$  est une forêt de  $n-p$  arcs, qu'elle est unique et minimale, ce qui démontre le théorème.

C.Q.F.D.

#### 4.A.4.3.2. Seconde variante - Cas des valuations pouvant être égales.

Soit  $G = (X, U, P_U)$  un graphe de  $p \geq 2$  "zones propres"  $C_i$  .

Si les arcs du graphe sont ordonnés suivant des valuations non décroissantes  $v(u_1) \leq v(u_2) \leq \dots \leq v(u_m)$  (\*), alors les  $p$  arbres obtenus par l'algorithme de forêt minimale constituent la forêt  $F = (X, A, P_A)$  graphe partiel de  $G = (X, U, P_U)$  de longueur minimale par rapport à toute autre forêt extraite de  $G$  .

(\*) P.ROSENSTIEHL, remarque 3 "Le cas des arêtes égales" [4 ]

La longueur de cette forêt est unique mais la forêt elle-même ne l'est pas.

F admet  $n-p$  arcs comme dans la précédente variante si la condition suffisante du § 4.D est satisfaite.

La forêt minimale extraite de  $G$  est toutefois unique dans la pratique puisqu'elle découle de l'ordre qui peut être imposé aux arcs de même longueur. En changeant l'ordre des arcs égaux on obtient une autre forêt dont la longueur minimale reste cependant invariante.

#### 4.B. VARIANTES DU PROBLEME FONDAMENTAL.

Dans la pratique l'existence d'ensembles  $C_i$ , tels que  $\text{Card}(C_i) > 1 \quad \forall i$ , entraîne celle de  $Z$  et vice-versa. Dans les problèmes de ce paragraphe on suppose que  $Z = \emptyset$ , ce qui implique donc que  $C = \bigcup_{i=1}^q C_i = \emptyset$ . Or on a vu que chaque ensemble  $C_i$  contient par hypothèse un sommet fixé  $c_i$  (\*). Sans perdre de vue l'aspect opérationnel des problèmes, on peut faire l'hypothèse suivante qui a l'avantage de permettre un passage plus aisé de ces problèmes à ceux des autres paragraphes (4.A. et 4.C.). On gagnera ainsi en généralité dans les problèmes de ce paragraphe 4.B.

La nouvelle hypothèse, qu'on gardera dorénavant, est que :

"L'ensemble  $Z$  ne sera jamais vide"

Alors deux cas se posent pour l'ensemble  $C$  (deux cas particuliers):

- 1)  $C$  n'existe pas ce qui est contradictoire avec le problème fondamental mais peut avoir d'autres applications (§4.B.1. Problème 1).
- 2)  $C$  existe et dans ce cas on peut avoir  $\text{Card}(C_i) = 1 \quad \forall i$ . On supposera alors que l'ensemble  $C_i$  est confondu avec le sommet fixé  $c_i$  (§4.B.2. Problème 2).

(\*) On donnera quelques précisions en notant que chaque  $c_i$  engendre, dans la pratique un ensemble  $C_i$  et un seul. L'ensemble  $C_i$  est contenu dans la "zone d'influence" engendrée par  $c_i$ . Le complémentaire de  $C_i$  appartient à la "zone de recouvrement"  $Z$ . On voit ainsi qu'une "zone d'influence" contient  $C_i$  plus une partie de la "zone de recouvrement".

On notera que cette nouvelle hypothèse, purement théorique, n'affecte en rien les résultats des algorithmes qui en découlent.

#### 4.B.1. Problème 1 - ( $C = \emptyset$ et $Z = X$ ).

Soit le graphe  $G = (X, U, P_U)$ .

Construire une forêt minimale de  $p$  composantes ( $p > 1$ ).

On donnera un graphe  $G$  de  $p_0$  composantes connexes ( $1 \leq p_0 \leq p$ )

- si  $p=1$ , on aura un arbre minimal de  $n-1$  arcs.

- si  $p_0 > p$  le problème est impossible.

- Il reste à étudier le cas  $1 \leq p_0 \leq p$  :

Données : Dans tous ces cas, les arcs sont ordonnés dans l'ordre non décroissant (\*). A chaque arc on associe ses deux extrémités dans la liste.

Procédure: Tout arc  $u_i$  qui est lu(\*\*) est retenu s'il ne crée pas de cycle avec ceux déjà sélectionnés (condition §4.A.4.). Si  $u_i$  crée un cycle, le rejeter et passer à l'arc  $u_{i+1}$  suivant, sinon retenir  $u_i$ . Il en est de même de  $u_{i+1}$  par rapport à  $u_{i+2}$  etc... Poursuivre la procédure jusqu'à sélection de  $n-p$  arcs.

Cette procédure ne diffère de celle de KRUSKAL I, exposée par P.ROSENSTIEHL [4], que par le fait que :

"le nombre des  $p$  composantes connexes de la forêt  $F_1$  graphe partiel obtenu ne reste pas forcément égal à celui du graphe initial  $G$  où  $p_0 \leq p$ .

Les composantes connexes obtenues forment les arbres de la forêt  $F_1$  minimale cherchée. Quand un arbre ne contient qu'un sommet unique sans boucle, on l'appelle arbre dégénéré. Il faut noter que tout arc non sélectionné engendrera un cycle ou un cocycle interdit

(\*) Il en sera de même pour les problèmes qui suivent dans ce chapitre.

(\*\*) dans la liste des arcs.

(propriété 2 du lemme 2) avec ceux de la forêt graphe partiel minimal  $F_1$ .

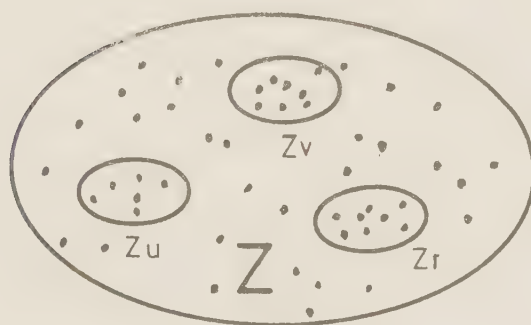
4.B.2. Problème 1 bis ( $C = \emptyset$  et  $Z = X$ ).

Soit le graphe  $G = (X, U, P_U)$  de  $p_0 \geq 1$  composante(s) connexe(s).

Construire une forêt minimale de  $p$  composantes ( $p > 1$ ).

On admettra l'existence d'ensembles  $Z_s$  de plus d'un élément.

Le problème se présente comme suit en prenant un graphe  $G$  connexe :



Légende  
 • Sommet de  $Z$   
 $Z_u, Z_v, Z_r \subset Z$

Figure 9 -  $\forall s \text{ Card}(Z_s) > 1$  avec  $Z_s \subset Z$ .

L'algorithme solution du problème 1 l'est également pour ce problème.

4.B.3. Problème 2 ( $\text{Card}(C_i) = 1$  et  $Z = X - C$ )<sup>(\*)</sup>

Soit le graphe  $G = (X, U, P_U)$  de  $p_0$  composantes connexes ( $p_0 \geq 1$ ).

Construire une forêt minimale de  $p$  composantes telle que  $q$  sommets fixés  $c_1, c_2, \dots, c_q$  appartiennent à  $q$  composantes distinctes ( $1 \leq p_0 \leq q \leq p$ ).

(\*) Si  $\text{card}(C_i) = 1$  alors le seul élément de la "zone propre"  $C_i$  est confondu avec le sommet fixé  $c_i$ .

- si  $1 = p_0 = q = p$  on aura un arbre minimal de  $n-1$  arcs obtenu par l'algorithme de Branch and Bound par exemple.
- si  $p_0 > p \Rightarrow$  problème impossible.
- si  $q > p$  la condition du lemme 1 ne peut être remplie puisqu'il existe forcément  $q-p$  chaînes entre les sommets fixés  $c_i$ ; ce qui est contraire à l'hypothèse de ce problème 2 où chaque sommet fixé  $c_i$  doit appartenir à une composante distincte.
- si  $q < p$  on aura  $p-q$  arbres minimaux sans sommet fixé  $c_i$ . Ce cas, quoique incompatible avec le problème fondamental, peut rentrer dans le cadre de ce qui reste à étudier du problème 2 ( $1 \leq p_0 \leq q < p$ ).
- Cas où  $1 \leq p_0 \leq q < p$  :

Données : la liste des arcs ordonnés dans l'ordre non décroissant.

Procédure: Tout arc lu dans cette liste est retenu :

- s'il remplit la condition  $\alpha$  (pas de cycle) ;
- s'il satisfait au lemme 1 (pas de chaîne entre deux sommets fixés  $c_i$ ).

Si  $u_i$  crée un cycle ou une chaîne entre deux sommets fixés  $c_i, c_j$  distincts, le rejeter et passer à l'arc  $u_{i+1}$ .

On recommence le processus jusqu'à sélection de  $n-p$  arcs, ce qui donnera la forêt  $F_2$ , graphe partiel minimal de  $G$ , dont certains arbres peuvent être dégénérés (sommet isolé).

#### 4.B.4. Problème 2 bis ( $\text{Card}(C_i) = 1 \ \forall i$ et $Z = X-C$ )

Soit le graphe  $G = (X, U, P_U)$  de  $p_0$  composante(s) connexe(s) ( $p_0 \geq 1$ ).

Construire une forêt minimale de  $p$  composantes telles que  $q$  sommets fixés  $c_1, \dots, c_q$  appartiennent à  $q$  composantes distinctes ( $1 \leq p_0 \leq q \leq p$ ).

| On admettra l'existence d'ensemble  $Z_s$  de plus d'un élément.

Le problème est schématisé comme suit si l'on prend un graphe connexe  $G$  :

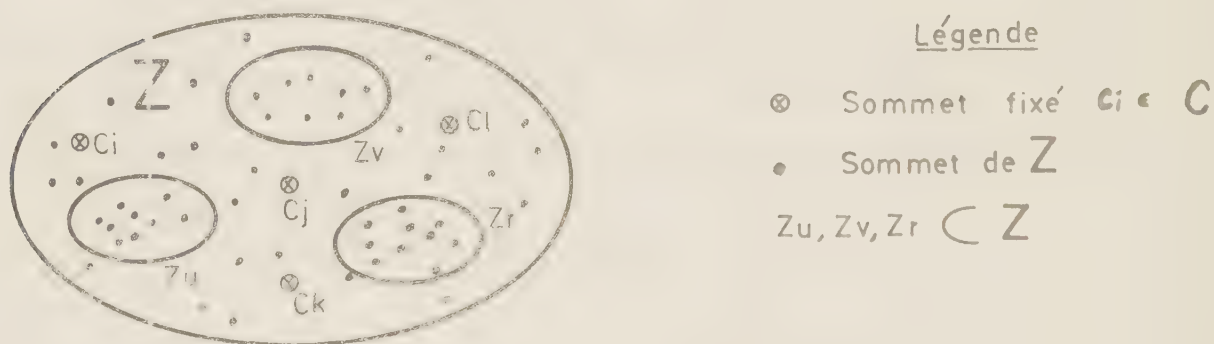


Figure 10 -  $\forall s \text{ Card}(Z_s) > 1 \text{ avec } Z_s \subset Z$ .

L'algorithme solution du problème 2 est aussi solution de ce problème.

4.B.5. Problème 2 ter ( $\text{card}(C_i) = 1 \forall i$  et  $Z = X - C$ ).

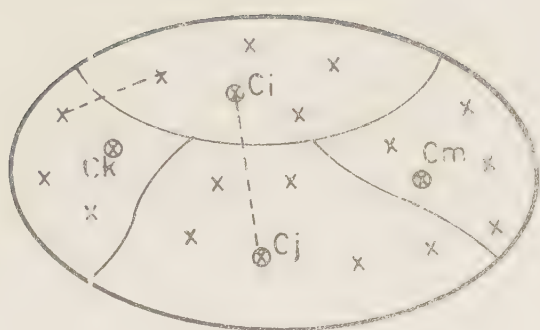
L'algorithme donnant la forêt minimale recherchée est le même que celui du problème 2.

Ce qui le diffère est que les  $p$  composantes de  $F_2$  bis, et les sommets que contient chacune d'elles, sont connus au préalable.

C'est un problème sans grand intérêt puisqu'il donne tout au plus la minimalité de la forêt vis-à-vis d'une décomposition imposée. Il ne s'agira donc pas du minimum absolu.

Ce problème peut être résolu par l'algorithme de Kruskal I si l'on part d'un graphe  $G = (X, U, P_U)$  déduit de  $G_0$  après élimination de tout arc créant une chaîne entre deux  $c_i$  distincts (lemme 1) (\*) ou reliant deux sommets de deux composantes distinctes de  $G$ . (Figure 11).

(\*) Dans les différents problèmes on part généralement du graphe  $G$  déduit de  $G_0$  (chap. IV, page 2, 4ème point) sauf pour le problème 1.



### Légende

⊗ Sommet fixé  $C_i$   
 Tel que  $C = \bigcup_{i=1}^p C_i$

x Sommet de  $\{X - C\}$

x---x } Arcs interdits à hôte  
 ⊗---⊗ } du graphe  $G_0$

Figure 11 - Graphe G

On vient de considérer le cas d'un graphe  $G$  de  $p_0 = 1$  composante connexe.

Si  $p_0 > 1$ , l'algorithme sera le même en notant toutefois que chacune des  $p$  composantes de  $F_2$  bis doit contenir un sommet fixé  $c_i$  et un seul.

### REMARQUE :

Le problème 1, ou tout autre problème issu du problème fondamental, peut présenter une variante dans laquelle les sommets de chacune des  $p$  composantes de  $F$  sont imposés comme faisant partie des données du problème.

On ne traitera pas ce genre de problèmes qui semblent présenter peu d'intérêt puisqu'ils ne donnent qu'une minimalité relative (\*) de la forêt graphe partiel du graphe  $G$ . On peut en revanche envisager une autre catégorie de problèmes (§4.C.) moyennant la nouvelle hypothèse suivante :

"L'ensemble  $C_i$  peut être partitionné en deux ou plusieurs sous ensembles"

(\*) Les problèmes qu'on a vu jusqu'ici donnent une minimalité absolue - aux contraintes près - : celles consistant à connecter un ensemble  $C_i$ , quand il existe, obligatoirement au sommet fixé  $c_i$  qui l'a engendré.

Cette hypothèse annule, dorénavant, celle posée dans l'introduction du chapitre IV selon laquelle chaque  $C_i$  se retrouve intégralement dans l'arbre correspondant de  $F$ .

#### 4.B.6. Problème fondamental - Seconde variante. (\*)

Soit le graphe  $G = (X, U, P_U)$  ayant  $p \geq 2$  "zones propres"  $C_i$ .

Construire une forêt minimale de  $p$  composantes connexes telle que :

- des ensembles de sommets,  $C_1, \dots, C_p$ , tous non vides appartiennent à  $p$  composantes distinctes;
- des sommets  $z_1, \dots, z_r$  soient connectés à l'un des ensembles  $C_1, \dots, C_p$  à l'exclusion de tous les autres;
- Le nombre de sommets de certains sous-ensembles de  $Z$  soit supérieur à un.

L'algorithme solution du problème fondamental sert également à résoudre ce problème.

#### 4.C. PROBLEMES POUVANT ADMETTRE LA PARTITION DES ENSEMBLES $C_i$ .

La procédure utilisée n'est pas forcément celle de construction (§4.A.3.) directe. L'ensemble  $C_i$  ne contient pas forcément de sommet fixé  $c_i$ .

##### 4.C.1. Problème 5.

Constuire à partir du graphe  $G = (X, U, P_U)$  une forêt  $F$  graphe partiel minimal de  $p$  composantes connexes telle que :

- $q$  ensembles de sommets  $C_1, \dots, C_q$  tous non vides appartiennent

(\*) C'est le problème 6 de la note (8) sur "les Algorithmes de Forêts minimales" qui se déduit du problème fondamental quand on admet l'existence des sous-ensembles de  $Z$  de plus d'un élément.

- à  $q'$  composantes distinctes avec  $q' \in [q, p]$  et où  $q \leq p$  ;
- les arbres de cette forêt peuvent contenir les sommets d'un seul ensemble  $C_i \forall i$ , en partie ou en totalité, à l'exclusion de tout sommet d'un autre  $C_i$ .

Le problème revient surtout à discuter les différents cas,  $q \leq q' \leq p$ , qu'on peut représenter schématiquement comme suit :

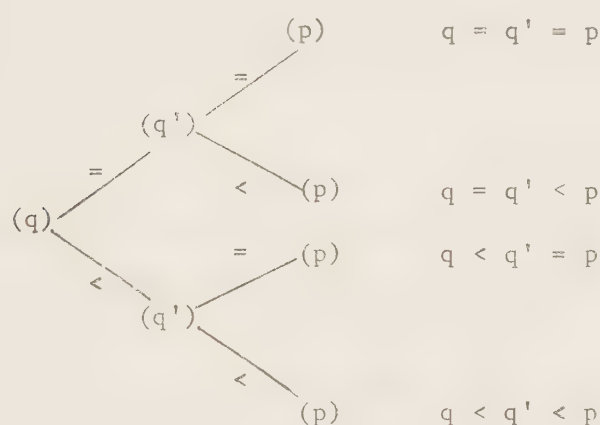


Figure 12 - Problèmes pouvant admettre la partition des ensembles  $C_i$ .

#### 4.C.1.1 - $q = q' = p$ .

C'est le problème 4 de la note (8) sur les "Algorithmes de Forêts minimales".

Il est résolu comme le problème 2 de ce chapitre avec cette différence qu'il s'agit de contraintes d'ensemble de sommets fixés  $C_1, \dots, C_q$  et non de simples sommets fixés isolés.

#### 4.C.1.2 - $q = q' < p$ .

Il existe  $p-q$  arbres minimaux de la forêt, ne contenant pas d'ensembles fixés  $C_i \forall i$  (\*), mais seulement

(\*) On admet que la partition des  $C_i$  n'a lieu que lorsque  $q' > q$ .

des sommets de  $Z$  .

4.C.1.3 -  $q < q' = p$  .

Il n'existe aucun arbre de  $F$  sans sommets de  $C$  puisque  $q' = p$  .

Cela implique que certains ensembles  $C_i$  sont partitionnés pour répondre à la seconde contrainte du problème. On partitionnera tout au plus  $q' - q$  ensembles  $C_i$  ou sous-ensembles de  $C_i \ \forall i$  .

4.C.1.4 -  $q < q' < p$  .

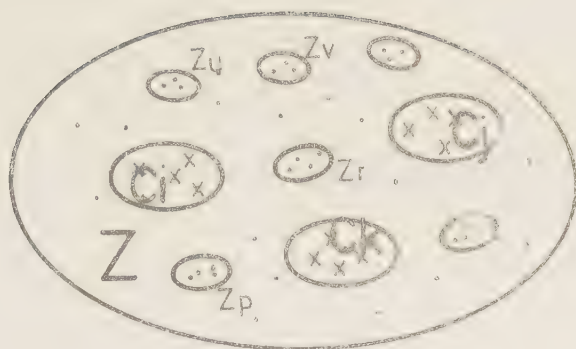
- Il existe  $p - q'$  arbres de la forêt minimale ne contenant aucun sommets de l'ensemble  $C = \bigcup_{i=1}^p C_i$  .
- On partitionnera au plus  $q' - q$  ensembles  $C_i$  ou sous-ensembles de  $C_i \ \forall i$  .

4.C.2. Problème 7 .

Construire à partir du graphe  $G = (X, U, P_U)$  une forêt  $F$  graphe partiel minimal de  $p$  composantes connexes telle que :

- $q$  ensembles de sommets  $C_1, \dots, C_q$  tous non vides appartiennent à  $q'$  composantes distinctes avec  $q' \in [q, p]$  et où  $q \leq p$  ;
- les arbres de cette forêt peuvent contenir les sommets d'un seul ensemble  $C_i \ \forall i$  , en partie ou en totalité , à l'exclusion de tout sommet d'un autre  $C_i$  .
- le nombre des sommets de certains sous-ensembles de  $Z$  soit supérieur à un .

Ce problème se résoud comme le précédent - problème 5 - . On le schématise comme suit en prenant un graphe  $G$  connexe :



### Légende

$C_i$  = Ensemble de sommets  
fixés avec  $C_i \subset C$

$Z_u, Z_v, Z_p, \dots \subset Z$   
"zone de recouvrement"

Figure 13

#### 4.D. CONDITION SUFFISANTE POUR QUE F ADMETTE $n-p$ ARCS ET SOIT SOLUTION DU PROBLEME 3 [8].

##### THEOREME :

Une condition suffisante pour que la forêt construite par l'algorithme (§ 4.A.4.1) soit solution du problème fondamental et admette  $n-p$  arcs est que dans  $G$  :

- (1) Tout sommet  $z_s \in Z$  est uni à au moins un sommet de  $C_{(s)}$  (\*)  
par un arc ou par une chaîne élémentaire dont tout sommet  $z_t$   
est tel que  $C_{(t)} \subset C_{(s)}$  ;
- (2) les  $p$  sous-graphes engendrés par les  $C_i$  sont connexes.

##### Démonstration

La condition (1) signifie qu'il n'existe pas de sommet  $z_s$ , ou d'ensembles de sommets de  $Z$ , qui ne soient liés, par un arc ou par une chaîne, à un  $C_i$  ( $i = 1, \dots, p$ ). Il existe donc des chaînes dont les arcs, liant tout sommet  $z_s$  à  $C_{(s)}$ , ne constituent pas de cocycle interdit tant que  $z_s$  n'est pas connecté à un des  $C_i \subset C_{(s)}$ . On voit que la condition (1) permet de connecter tout sommet  $z_s$  à l'un des sous-graphes connexes, et à un seul, engendrés par les  $C_{(s)}$ .

Par ailleurs dans les  $C_i$  ( $i = 1, \dots, p$ ), tous connexes, tout arc ne créant pas de cycle  $F$  est retenu.

(\*)  $C_{(s)}$  est l'ensemble des sommets de  $C$  pouvant être connectés à  $z_s$  tel que  $\forall s, \exists i$  tel que  $C_i \subset C_{(s)}$ .

Il en est de même de tout arc liant  $z_s$  à  $C_i \subset C_{(s)}$  tant que  $z_s$  n'est pas connecté à  $C_j \subset C_{(s)}$  ( $C_j \neq C_i$ ). On obtient ainsi la formation dans  $F$  d'un sous-graphe connexe incluant  $C_i$  et éventuellement des sommets  $z_s$ .

On notera que la suppression de certains sommets  $z_s$  peut disconnecter la forêt optimale mais sans disconnecter dans  $G$  le **sous**-graphe engendré par les  $C_i$  ( $C$  et  $Z$  partitionnent  $X$ ).

Par conséquent, tout  $z_s$  est rattachable dans la forêt à un  $C_i$ , et à un seul. Tout  $C_i$ , y compris les  $z_s$  qui lui sont connectés, engendre un sous-graphe connexe et on peut sélectionner  $n-p$  arcs. Le dernier arc sélectionné connectera deux sommets d'un même  $C_i$  ou un sommet  $z_s$  à un sommet de  $C_j \subset C_{(s)}$  ou un sommet  $z_s$  à un autre sommet  $z_t$  tel que  $C_{(t)} \subset C_{(s)}$ .

#### CONDITION SUFFISANTE (1) NON SATISFAITE

- |        |                                                     |
|--------|-----------------------------------------------------|
| Cas 1) | → on trouve cependant la solution par l'algorithme, |
| Cas 2) | → l'algorithme ne conduit pas à la solution.        |

On a (figures 14 et 15) :

- $C_{(1)} = C_1 \cup C_2$   
 $C_{(2)} = C_1 \cup C_4$   
 $C_{(3)} = C_3 \cup C_4$
- le sommet  $z_2$  qui n'est pas uni à un sommet de  $C_{(2)}$  par un arc ou par une chaîne élémentaire dont tout sommet  $z_t$  est tel que  $C_{(t)} \subset C_{(2)}$ .

Cas 1) -  $p=4$  FORET DE 4 COMPOSANTES (convergence)

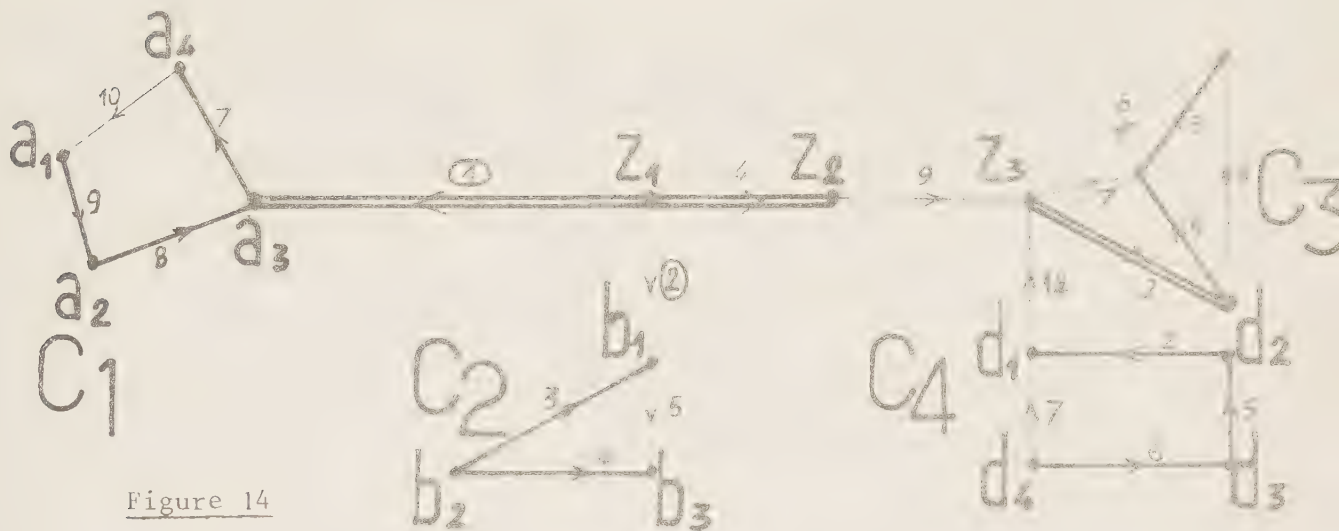


Figure 14

Cas 2) -  $p=4$  FORET DE 5 COMPOSANTES ( $z_2$  reste isolé)

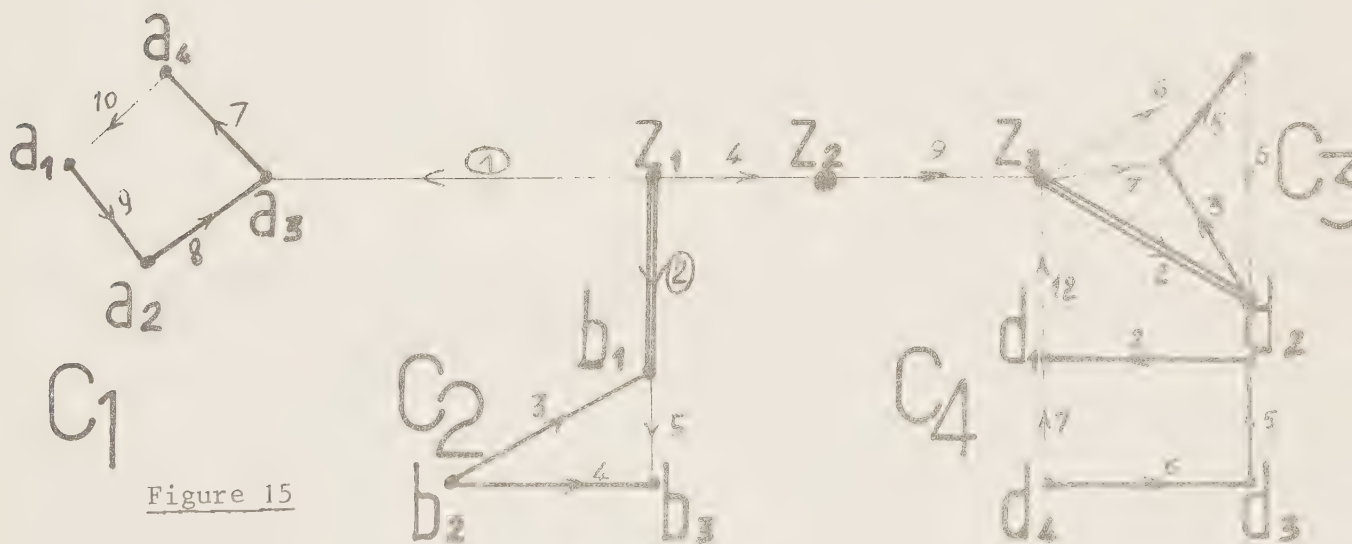


Figure 15

$z_s$  reste isolé suite aux nouvelles hypothèses faites sur les longueurs des arcs [2 et 1 au lieu de 1 et 2 respectivement sur les longueurs des arcs  $(a_3, z_1)$  et  $(z_1, b_1)$ ].

Le sommet  $z_2$  ne peut être connecté qu'à  $C_1$  ou  $C_4$  ( $C(2) = C_1 \cup C_4$ ). On ne peut donc retenir dans  $F$  les arcs  $(z_1, z_2)$  ou  $(z_2, z_3)$  sinon

on connecte  $z_2$  à  $C_2$  ou à  $C_3$  ce qui est contraire à l'hypothèse  $C_{(2)} = C_1 \cup C_4$ .

La forêt solution du cas 1) est encore optimale pour le cas 2) avec un supplément de longueur de 1) mais l'algorithme (§ 4.A.4.1) ne permet pas de l'obtenir. On aura 5 composantes, dont un arbre dégénéré, au lieu de 4.

#### CONDITION SUFFISANTE (2) NON SATISFAITE

Cas 1) → on trouve cependant la solution par l'algorithme.  
 Cas 2) → l'algorithme ne conduit pas à la solution.

Si les  $p$  sous-graphes engendrés par les  $C_i$  ne sont pas connexes dans  $G$  [figures 16 et 17 où  $b_3$  n'est lié aux autres sommets de l'ensemble  $C_b = \{b_1, b_2, b_3, b_4, b_5\}$  qu'à travers le sommet  $z_1 \in Z$ ] sans passer par un sommet de  $Z$  alors on a :

Cas 1) -  $p=2$  - FORET A 2 COMPOSANTES (convergence)

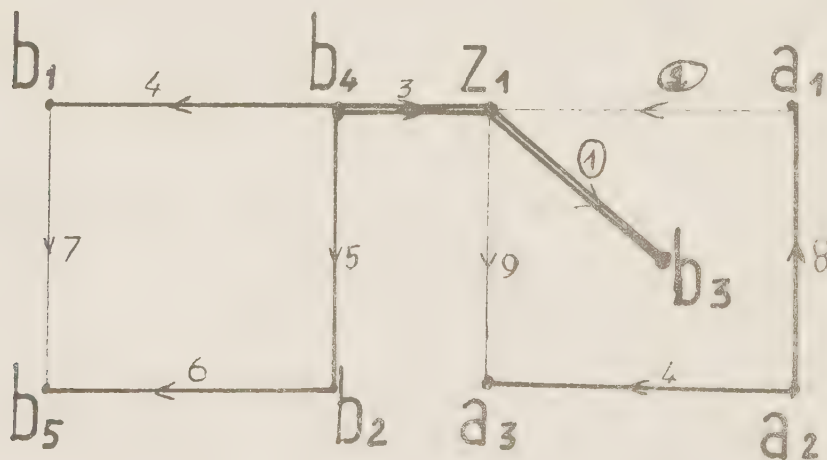


Figure 16

Cas 2) -  $p=2$ 

FORET A 3 COMPOSANTES (bloccage)

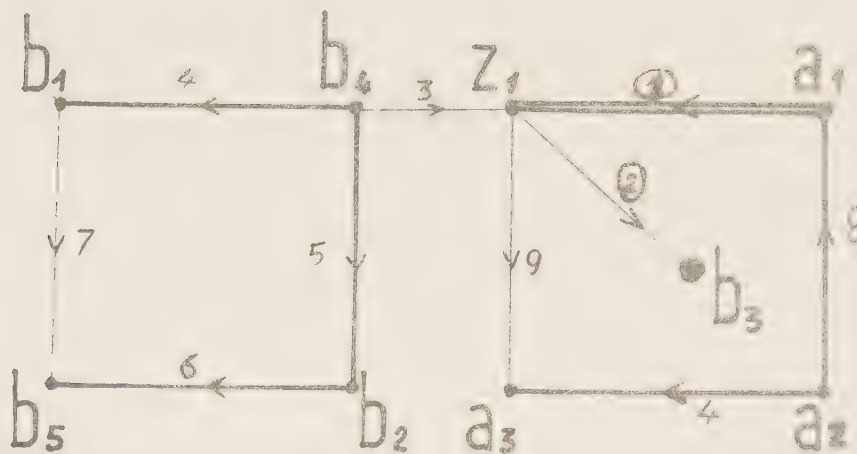


Figure 17

La forêt ( $p=2$ ) solution du cas 1) est aussi solution du cas 2) mais l'algorithme (§ 4.A.4.1) ne permet pas de l'obtenir suite aux nouvelles hypothèses faites sur les longueurs des arcs [2 et 1 au lieu de 1 et 2 respectivement sur les arcs  $(z_1, b_3)$  et  $(z_1, a_1)$ ]. Cette solution, si elle était obtenue, aurait la même longueur (à 1 près par excès) que dans le cas 1). Le premier arc sélectionné dans  $F$ , de longueur 1, connectera  $z_1$  à  $C_b$  ou à  $C_a = \{a_1, a_2, a_3\}$  respectivement dans les figures 16 et 17. Dans le cas où  $z_1$  est lié à  $C_a$ , le sommet  $b_3$ , qui ne peut être connecté à  $C_b$  qu'à travers  $z_1$ , restera isolé ( $C_b$  n'engendre plus de graphe connexe dans  $F$ ).

#### 4.E. CONCLUSION

• Sur le plan application des résultats trouvés, on notera que l'algorithme de forêt minimale vise, dans le domaine postale, davantage le découpage du territoire en régions postales, du point de vue acheminement du courrier, que l'organisation optimale des réseaux de transport qui en découle et qui est un sous-produit non moins négligeable.

Cette notion du "meilleur choix" ou de la "meilleure affectation" d'un ensemble de sommets susceptibles de "rattachement" à plusieurs ensembles distincts trouve toute son importance dans de nombreux domaines (chapitre I - Introduction) où le choix d'une stratégie , d'une option d'organisation, s'impose avant même le lancement de tout processus complexes et très onéreux.

Comme autre type d'application on remarquera aussi celui de la "représentation d'un ensemble de mots" [12] par J.C.DERNIAME et C.PAIR , à l'aide d'une forêt, dans les "problèmes de recherche de chemins".

• On vient de voir dans ce chapitre la généralisation du problème de l'arbre minimal à celui d'une forêt minimale.

On est parti pour cela de l'algorithme Kruskal I tel qu'il est présenté par Rosenstiehl [4].

Peut-on à partir de Kruskal II , dual de Kruskal I , résoudre quelques uns des problèmes de ce chapitre ou bien leur totalité ? On sait toutefois que la problème 1 (§ 4.B.1) a déjà été résolu par Rosenstiehl (Kruskal II).

Il restera à étudier :

- les autres problèmes (Problèmes 2, 4, 5, 6, 7 et le fondamental) à partir de Kruskal II ;
- puis l'ensemble des problèmes, y compris le problème 1 , à partir de Kruskal III et Kruskal IV pour voir dans quelle mesure ces trois algorithmes sont généralisables ou non.

## CHAPITRE V

### APPLICATIONS POSTALES<sup>(\*)</sup>

#### 5.1. INTRODUCTION.

L'organisation postale est un domaine dans lequel on inaugure, il me semble, la théorie des graphes au sens le plus formel du mot; c'est-à-dire celui de la Combinatoire où l'on étudie, de façon structurée, toutes les configurations d'un graphe donné, dans le cadre des mathématiques modernes.

Une des premières préoccupations de mes activités au sein des équipes spécialisées, probablement la plus importante, est de concilier l'esprit de rigueur et d'exactitude de la Mathématique avec celui de rentabilité industrielle. On a parfois tendance à préférer une "bonne heuristique" à un algorithme mathématique, c'est-à-dire exact, et cela pour plusieurs raisons dont voici quelques unes :

- Respect du délai imparti à l'étude, délai d'ailleurs souvent très réduit (rentabilité industrielle);
- Impératifs relevant des décisions non programmables, c'est-à-dire d'ordre socio-administratif (ces impératifs peuvent être moindre dans les pays où l'organisation postale est encore embryonnaire);
- L'Aléatoire joue un rôle très important dans les prévisions postales.

---

(\*) Ce chapitre s'adresse particulièrement à ceux qui s'intéressent aux applications mathématiques ou à l'organisation postale.

Pour ces diverses raisons entre autres on recherche avant tout des solutions fonctionnelles, c'est-à-dire qui répondent aux contraintes du problème sans être optimales. La recherche de l'optimalité sera alors l'idéal vers lequel on tend au fur et à mesure des nécessités.

Le système postal peut être vu comme un ensemble de sous-systèmes en interaction permanente. Ainsi par exemple l'automatisation d'une région postale donnée (\*) ne peut se faire indépendamment des contraintes des autres régions du territoire. L'équipement d'un Centre de tri n'est pas seulement fonction du courrier qu'il engendre dans la région qu'il couvre, mais également du courrier qu'il reçoit des autres G.A.P. et de l'extérieur du territoire.

Au niveau exploitation de ce système, une fois que les structures et les équipements sont mis en place, on pourrait intégrer, dans un futur plus ou moins proche, la mise en place d'un processus de régulation automatique dans lequel des calculateurs reliés au calculateur principal permettraient une meilleure gestion du système au niveau du territoire. Cette "meilleure gestion" signifie une meilleure utilisation de l'équipement existant en évitant, dans les prévisions au cours de l'évolution du système, des sous-équipements et des sur-équipements des centres de traitement du courrier. Elle signifie surtout le respect de la "qualité de service" (\*\*): c'est-à-dire le temps mis par une correspondance (\*\*\*) entre la dernière levée utile (\*\*\*\*) du point où se trouve l'expéditeur et sa remise au destinataire reste constant (\*\*\*\*\*), quelque soit les lieux, au niveau du territoire, des deux correspondants.

L'organisation postale est un de ces domaines, qui pour mener à bien les missions qu'on se propose, nécessite puisqu'il s'agit d'une organisation récente, donc sans passé technique, énormément de prudence dans la prise des décisions, d'autant plus qu'il s'agit d'un

---

(\*) appelée G.A.P. (Groupement d'Acheminement Postal).

(\*\*) les définitions proposées des termes postaux sont données à titre indicatif, il ne s'agit donc pas de définitions codifiées.

(\*\*\*) courrier urgent (ou courrier ordinaire) et courrier non urgent (ou courrier long).

(\*\*\*\*) dernier ramassage de la journée du courrier à un point donné.

(\*\*\*\*\* ) on compte une journée pour le courrier ordinaire et trois journées pour le courrier long. Ce délai peut varier d'un pays

domaine où l'Aléatoire joue un rôle très important. Les personnes que j'ai rencontrées et qui sont chargées de la préparation de l'ensemble des décisions<sup>(\*)</sup> sont de très haut niveau technique et administratif et ont une parfaite connaissance du mécanisme postal pour avoir vécu le plus souvent de nombreux problèmes que pose un tel processus probabiliste.

Ma modeste contribution depuis octobre 1970 au cours de nombreuses réunions tenues à la DIRECTION GENERALE DES POSTES à Paris et surtout à SOMEPOST à Orly se traduit par la formalisation des problèmes posés en vue d'aboutir à un ensemble de décisions programmables. Il s'agit donc de ramener les études au problème de la Recherche Opérationnelle dans laquelle la théorie des graphes trouve un épanouissement idéal.

On traitera ci-après quelques problèmes d'organisation postale de manière tout à fait brève; le but étant de montrer l'élégance des solutions, facilement manipulables, obtenues par l'utilisation de la théorie des graphes. On s'arrêtera donc au niveau "principes de base" en ne présentant que quelques problèmes parmi ceux qu'on avait à résoudre.

On verra successivement :

- des exemples d'application de l'arbre minimal et de la forêt minimale;
- une procédure exhaustive qu'il apparaissait nécessaire d'établir pour vérifier les résultats de l'algorithme de forêt minimale;
- l'alimentation fonctionnelle d'un Centre de tri (lois d'arrivées du courrier au Centre).
- et enfin la détermination de la zone d'influence d'un Centre de tri en vue de procéder au découpage du territoire en régions postales.

---

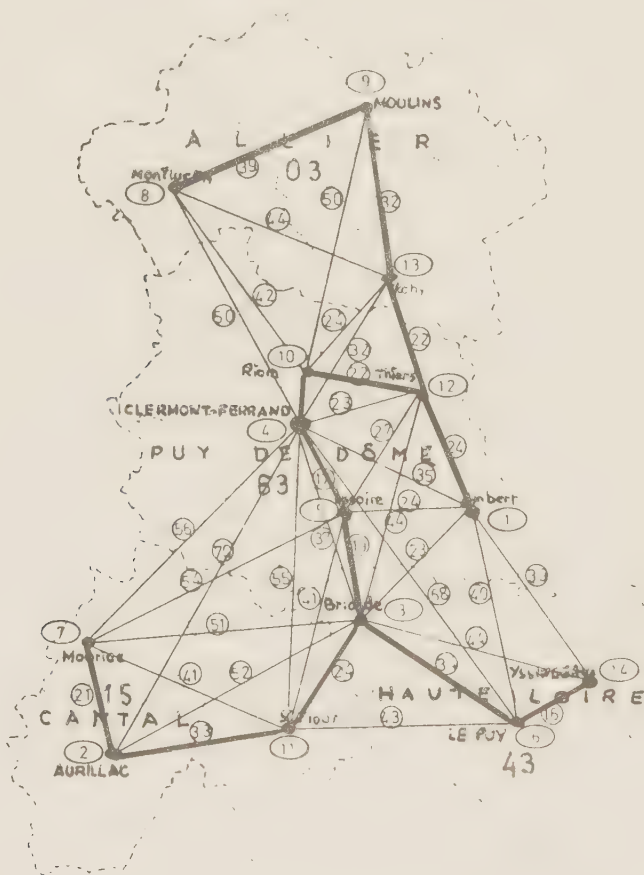
(\*) et du choix de la meilleure d'entre elles.

## 5.2. EXEMPLES D'APPLICATION DE L'ARBRE MINIMAL ET DE LA FORET MINIMALE.

### 5.2.1. Problème A : Arbre minimal - Réseau d'acheminement du courrier de la Région Postale de Clermont-Ferrand. [ 11 ].

Il s'agit à partir du graphe  $G = (X, U, P_U)$ , de la figure 18 ci-dessous, de déterminer un réseau d'acheminement du courrier de la Région Postale de CLERMONT-FERRAND, supposée contenir quatre départements (\*), dont le coût d'exploitation soit minimal.

Figure 18



Chaque bureau de poste est représenté par un point, les liaisons entre bureaux sont indiquées symboliquement par des droites. Le Centre de tri est considéré comme un point quelconque (il s'agit de (4) celui de Clermont-Ferrand). Les nombres encadrés représentent les valeurs des liaisons à vol d'oiseau exprimées en mm.

(\*) Les départements du CANTAL, de la HAUTE-LOIRE, du PUY DE DOME et de l'ALLIER.

Soit le graphe  $G = (X, U, P_U)$  précédemment défini, où :

- $X$  représente l'ensemble des préfectures et sous-préfectures (sommets);
- $U$  indique l'ensemble des liaisons (arcs) entre deux sommets quelconques;
- $P_U$  est la fonction de valuation des arcs.

Il s'agit d'extraire de ce graphe une configuration composée de sommets et d'arcs qui soit :

- connexe de façon à permettre à deux sommets quelconques, dans le graphe  $G$ , d'échanger du courrier entre eux ;
- sans cycles ; l'existence d'un cycle signifie celle d'un arc superflu puisque déjà tous les sommets peuvent être atteints de n'importe lequel d'entre eux.
- et optimale ; il s'agit de la minimalité du coût de transport. C'est-à-dire disposant d'un ensemble de moyens de transport<sup>(\*)</sup> et d'une organisation spécifiant l'utilisation de ces moyens en fonction de la quantité du courrier, du temps, du type de voiture etc..., on doit parcourir une distance globale minimale au niveau régional.

La configuration cherchée est celle de l'arbre minimale (traits plus gros de la figure 18).

#### CLERMONT FERRAND

|            |               |                |              |
|------------|---------------|----------------|--------------|
| 01 AMBERT  | 02 AURILLAC   | 03 BRIEUE      | 04 CLERMONT  |
| 05 ISSOIRE | 06 LE PUY     | 07 MAURIAC     | 08 MONTLUCON |
| 09 MOUTINS | 10 RIOM       | 11 SAINT-FLOUR | 12 THIERS    |
| 13 VICHY   | 14 YSSINGEAUX |                |              |

| N° | RESEAU OPTIMAL |               |
|----|----------------|---------------|
|    | DISTANCES      | POINTS RELIES |
| 1  | 24             | 1 5           |
| 2  | 17             | 5 4           |
| 3  | 10             | 4 10          |
| 4  | 19             | 5 3           |
| 5  | 22             | 10 12         |
| 6  | 22             | 12 13         |
| 7  | 24             | 3 11          |
| 8  | 32             | 13 9          |
| 9  | 33             | 11 2          |
| 10 | 21             | 2 7           |
| 11 | 34             | 3 6           |
| 12 | 16             | 6 14          |
| 13 | 39             | 9 8           |

(\*) voitures par exemple.

### 5.2.2. Problème B :

Déterminer le meilleur rattachement d'un quanta postal (\*) au Centre de Tri le plus intéressant au point de vue économique.

De l'idée que le sous-graphe partiel minimal du graphe  $G = (X, U, P_U)$  n'est pas forcément le même selon que le sous-graphe est contenu dans  $G = (X, U, P_U)$  ou dans  $V = (X', U', P_{U'})$  avec  $G \neq V$ , on déduit qu'il peut exister deux sous-graphes partiels  $H_1 = (X_1, U_1, P_{U_1}) \subset G$  et  $H_2 = (X_2, U_2, P_{U_2}) \subset V$  où :

longueur  $(H_1) \leq$  longueur  $(H_2)$  , sinon  
longueur  $(H_2) >$  longueur  $(H_1)$  .

Il s'agit de la longueur somme de tous les arcs de  $H_1$  (soit  $S_1$ ) et de  $H_2$  (soit  $S_2$ ).

Or on a vu par ailleurs au problème A , que le coût de transport augmente en fonction de la distance parcourue.

On en déduit que le rattachement d'une zone postale à un Centre de Tri plutôt qu'à un autre est "meilleur" selon que le réseau d'acheminement du courrier est plus court dans le premier cas que dans le second.

Rappelons qu'il s'agit de comparer les sous-arbres minimaux extraits d'un "même sous-graphe" , c'est-à-dire de  $H_1 \subset G$  et de  $H_2 \subset V$  ( $X_1 = X_2$  ,  $U_1 = U_2$  et  $P_{U_1} = P_{U_2}$ ). On prendra ensuite le sous-arbre minimal de plus courte longueur.

#### 5.2.2.1 - Procédure d'application .

- Appliquer l'algorithme BBB par exemple, donnant l'arbre minimal, à la région postale N°1 (cf. figure 19). Cette région contient une "zone de recouvrement" avec la région postale N°2.
- Opérer de même pour la région postale N°2.
- Noter dans les deux cas la somme des longueurs des arcs du réseau d'acheminement du courrier de la zone de recouvrement et soient

(\*) On a pris l'arrondissement comme quantum postal. Dans ce problème il s'agit du rattachement global d'un quanta postal. On verra dans le problème de la forêt minimale que le rattachement est plus fin puisqu'il se fait au niveau quantum postal, et est en outre optimal.

$S_1$  et  $S_2$  ces deux sommes.

- Si  $S_1 > S_2$ , connecter la "zone de recouvrement" au Centre de tri engendrant  $S_2$ .
- Si  $S_1 < S_2$ , faire l'inverse.
- Si  $S_1 = S_2$ , rattachement indifférent.

#### 5.2.2.2 - Résultats.

On se rapportera à la figure 19 et au listing suivant dans lequel le coût de rattachement est égal à la différence entre le coût des liaisons de l'arbre minimal du GAP avec zone puis du GAP sans zone. Le rattachement est en faveur du Centre de Clermont-Ferrand puisqu'on enregistre un gain de 20.

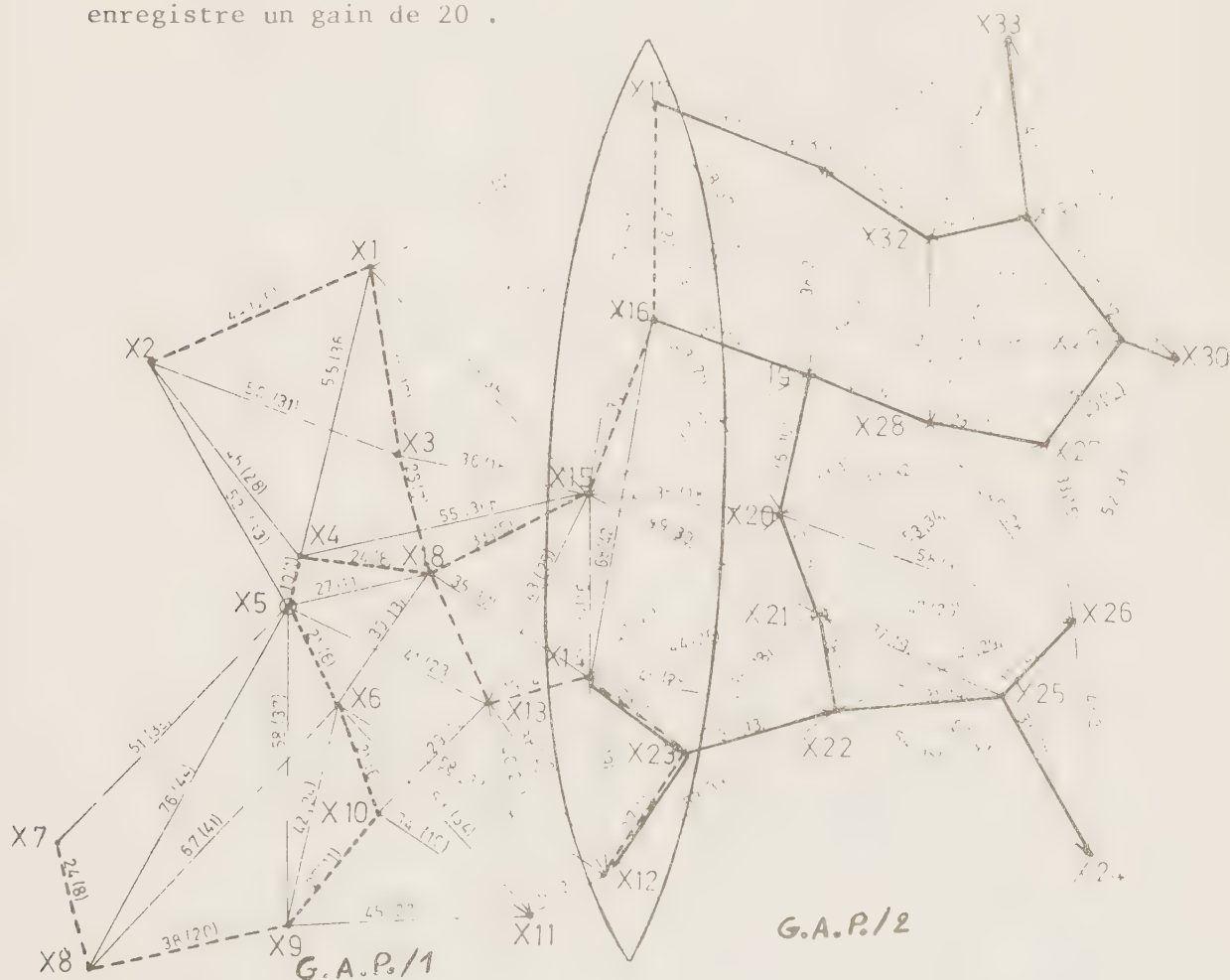


Figure 19 : Graphe des liaisons possibles des G.A.P./1 et G.A.P./2

MEILLEUR RATTACHEMENT D'UN QUANTA POSTAL

GAP 1:CLERMONT

|                  |          |    |    |
|------------------|----------|----|----|
| LIAISON INTERNE  | * 1* 3** | 35 | ** |
| LIAISON INTERNE  | * 3*18** | 23 | ** |
| LIAISON INTERNE  | *18* 4** | 24 | ** |
| LIAISON INTERNE  | * 4* 5** | 10 | ** |
| LIAISON INTERNE  | * 5* 6** | 21 | ** |
| LIAISON INTERNE  | * 6*10** | 21 | ** |
| LIAISON INTERNE  | *18*13** | 26 | ** |
| LIAISON DE RATT. | *13*14** | 19 | ** |
| LIAISON EXTERNE  | *14*23** | 23 | ** |
| LIAISON INTERNE  | *10* 9** | 27 | ** |
| LIAISON EXTERNE  | *23*12** | 27 | ** |
| LIAISON DE RATT. | *12*11** | 16 | ** |
| LIAISON DE RATT. | *18*15** | 33 | ** |
| LIAISON EXTERNE  | *15*16** | 34 | ** |
| LIAISON INTERNE  | * 9* 8** | 38 | ** |
| LIAISON INTERNE  | * 8* 7** | 24 | ** |
| LIAISON EXTERNE  | *16*17** | 40 | ** |
| LIAISON INTERNE  | * 1* 2** | 44 | ** |

COUT DE , GAP 1 AVEC ZONE: 485

|                 |          |    |    |
|-----------------|----------|----|----|
| LIAISON INTERNE | * 1* 3** | 35 | ** |
| LIAISON INTERNE | * 3*18** | 23 | ** |
| LIAISON INTERNE | *18* 4** | 24 | ** |
| LIAISON INTERNE | * 4* 5** | 10 | ** |
| LIAISON INTERNE | * 5* 6** | 21 | ** |
| LIAISON INTERNE | * 6*10** | 21 | ** |
| LIAISON INTERNE | *18*13** | 26 | ** |
| LIAISON INTERNE | *10* 9** | 27 | ** |
| LIAISON INTERNE | *10*11** | 34 | ** |
| LIAISON INTERNE | * 9* 8** | 38 | ** |
| LIAISON INTERNE | * 8* 7** | 24 | ** |
| LIAISON INTERNE | * 1* 2** | 44 | ** |

COUT DE , GAP 1 SANS ZONE: 327  
COUT DE RATTACHEMENT: 158

## GAP 2:LYON

|                  |          |    |    |
|------------------|----------|----|----|
| LIAISON INTERNE  | *19*28** | 25 | ** |
| LIAISON INTERNE  | *28*27** | 23 | ** |
| LIAISON INTERNE  | *27*29** | 24 | ** |
| LIAISON INTERNE  | *29*30** | 10 | ** |
| LIAISON INTERNE  | *19*20** | 26 | ** |
| LIAISON INTERNE  | *20*21** | 20 | ** |
| LIAISON INTERNE  | *21*22** | 18 | ** |
| LIAISON INTERNE  | *29*31** | 29 | ** |
| LIAISON INTERNE  | *31*32** | 18 | ** |
| LIAISON INTERNE  | *32*34** | 24 | ** |
| LIAISON DE RATT. | *22*23** | 30 | ** |
| LIAISON EXTERNE  | *23*14** | 23 | ** |
| LIAISON EXTERNE  | *23*12** | 27 | ** |
| LIAISON DE RATT. | *19*16** | 31 | ** |
| LIAISON INTERNE  | *22*25** | 31 | ** |
| LIAISON INTERNE  | *25*26** | 19 | ** |
| LIAISON DE RATT. | *34*17** | 33 | ** |
| LIAISON INTERNE  | *25*24** | 33 | ** |
| LIAISON INTERNE  | *31*33** | 34 | ** |
| LIAISON EXTERNE  | *14*15** | 34 | ** |

COUT DE , GAP 2 AVEC ZONE: 512

|                 |          |    |    |
|-----------------|----------|----|----|
| LIAISON INTERNE | *19*28** | 25 | ** |
| LIAISON INTERNE | *28*27** | 23 | ** |
| LIAISON INTERNE | *27*29** | 24 | ** |
| LIAISON INTERNE | *29*30** | 10 | ** |
| LIAISON INTERNE | *19*20** | 26 | ** |
| LIAISON INTERNE | *20*21** | 20 | ** |
| LIAISON INTERNE | *21*22** | 18 | ** |
| LIAISON INTERNE | *29*31** | 29 | ** |
| LIAISON INTERNE | *31*32** | 18 | ** |
| LIAISON INTERNE | *32*34** | 24 | ** |
| LIAISON INTERNE | *22*25** | 31 | ** |
| LIAISON INTERNE | *25*26** | 19 | ** |
| LIAISON INTERNE | *25*24** | 33 | ** |
| LIAISON INTERNE | *31*33** | 34 | ** |

COUT DE , GAP 2 SANS ZONE: 334

COUT DE RATTACHEMENT: 178

CLERMONT , GAIN: 20

5.2.4. Problème C : Forêt minimale - Découpage du Territoire en Régions Postales sur le plan du Réseau d'acheminement du courrier.

( cf programmes AB140P et AB141P) [11] et leurs résultats au chapitre 6 .

ENONCE :

Soit un ensemble de Centres de tri. Chacun d'eux peut traiter le courrier d'une région postale<sup>(\*)</sup>. Or il existe des cas où les bureaux de certaines zones postales, appelées alors "zones de recouvrement" ont leur courrier qui peut être traité par deux ou plusieurs Centre de Tri<sup>(\*\*)</sup> étant donné qu'ils répondent aux critères imposés par ces Centres. On demande alors de construire :

- Une procédure mathématique donnant la répartition optimale des bureaux de toutes les zones de recouvrement;
- l'ensemble des réseaux de transport optimaux au niveau du territoire.

Il s'agit du problème de la forêt minimale qui a été longuement traité aux chapitres 1 et 4 .

On sera donc bref ici en se limitant à un exemple concret : celui des régions postales des Centres de tri de Clermont-Ferrand et de Lyon comprenant respectivement les départements suivants :

- L' ALLIER, le PUY-DE-DOME, le CANTAL, la HAUTE-LOIRE<sup>(\*\*\*)</sup> , la LOIRE et la SAONE-ET-LOIRE pour la première région;
- La HAUTE-LOIRE , la LOIRE , la SAONE-ET-LOIRE , le RHONE , l'AIN et l'ISERE pour la seconde.

On ne prendra que deux régions, au lieu de l'ensemble du territoire, car il n'est pas nécessaire, au stade actuel<sup>(\*\*\*\*)</sup> d'utiliser des moyens de calculs plus puissants.

Le problème consiste surtout à affecter les bureaux des départements soulignés, qui se trouvent dans la "zone d'influence" des deux Centres

---

(\*) (cf problème A). Prenons deux ou trois départements pour fixer les idées. La superficie étant en effet fonction de la densité démographique et des moyens mis en oeuvre (équipements ...)

(\*\*) Chaque bureau de poste doit se trouver dans au moins une zone d'influence. On suppose par ailleurs qu'aucun Centre de tri ne se trouve dans la zone d'influence d'un autre Centre.

(\*\*\*) les départements soulignés sont en commun dans les deux régions;

(\*\*\*\*) celui de la construction mathématique.

de Tri soit à Clermont-Ferrand soit à Lyon. C'est-à-dire qu'à partir du graphe  $G = (X, U, P_U)$  (Fig.20) où

X12 YSSINGEAUX  
 X14 MONTBRISSON  
 X15 ROANNE  
 X16 CHAROLLES  
 X17 AUTUN  
 X23 SAINT-ETIENNE

appartiennent à la "zone de recouvrement", on déduit, après application de l'algorithme de la forêt minimale, la forêt graphe partiel minimal (figure 21).

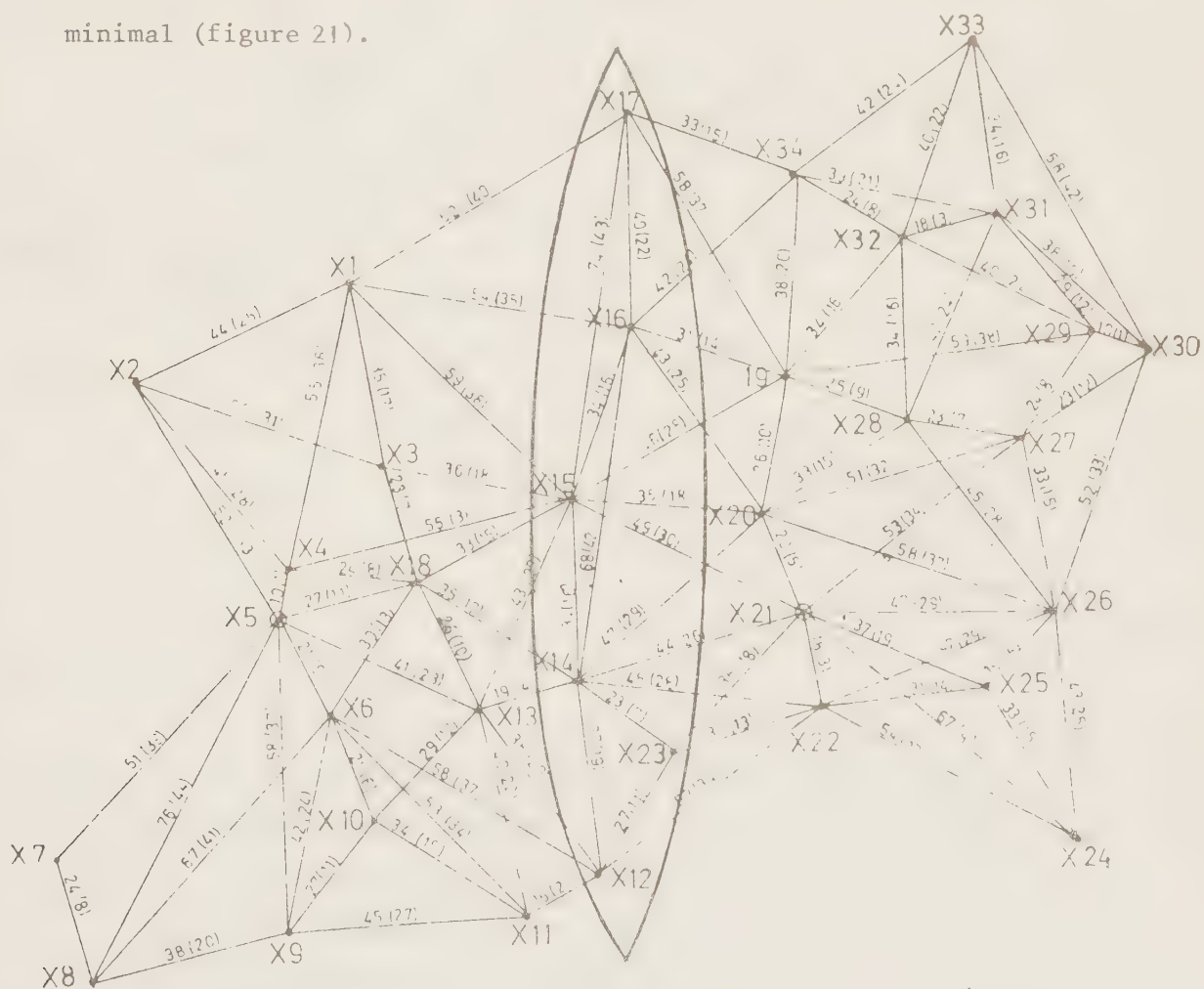


Figure 20 : Graphe des liaisons possibles des Régions

Postales de Clermont-Ferrand (X5) et de LYON (X21).

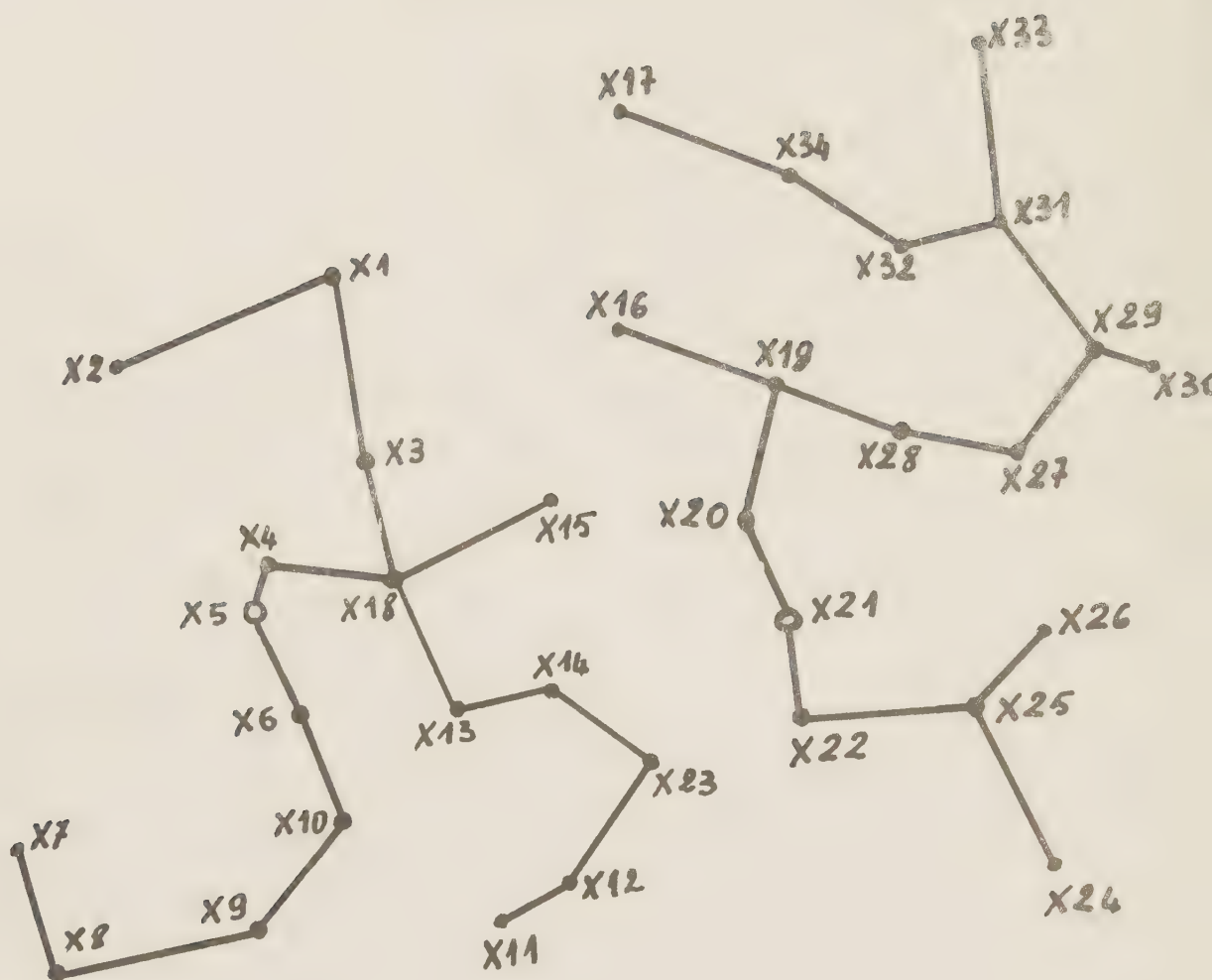


Figure 21 : Forêt minimale .

A titre indicatif on donnera la figure 22 où l'on remarque que les arbres minimaux des deux régions ont par rapport à la forêt graphe partiel minimale, les liaisons (X15,X16) , (X16,X17) en plus et celles (X14,X23) , (X12,X23) en double .

La suppression de ces liaisons montre l'intérêt du modèle mathématique qu'on a élaboré , en particulier sur le plan coût d'exploitation du réseau d'acheminement du courrier.

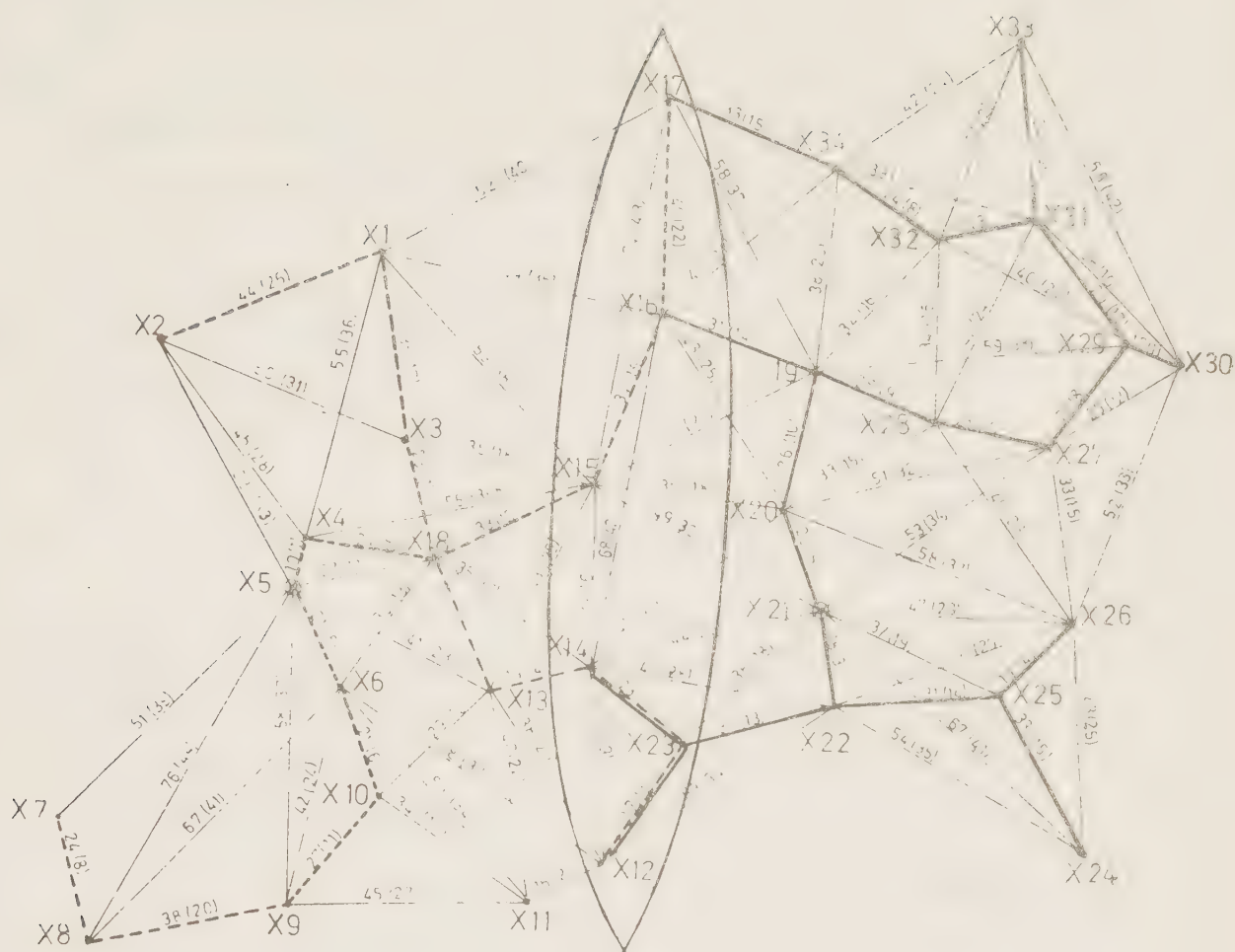


Figure 22 : Les arbres minimaux sont en pointillé pour la région de Clermont-Ferrand et en traits renforcés pour celle de Lyon

#### 5.2.3.1 - Remarque 1 :

La recherche du chemin hamiltonien est exclu dans ce problème quoiqu'il réduise le nombre de moyens de transport à l'unité car l'alimentation du Centre de Tri est une loi discrète (à fréquence et à amplitude variables).

#### 5.2.3.2 - Remarque 2 :

L'unique Centre de traitement du courrier (appelé Centre de Tri) se trouve à Clermont-Ferrand. D'une manière

générale la distribution du courrier se fait le matin et la collecte le soir. Dans certains points on peut avoir plusieurs distributions ou collectes par jour.

#### 5.2.3.3 - Programmation.

On notera que chacune des deux méthodes, BBB ou Kruskal I, est mieux adaptée que l'autre pour résoudre certains types de problèmes d'organisation postale.

### 5.3. PROBLEME C BIS : FORET MINIMALE - PROCEDURE EXHAUSTIVE.

Cette procédure n'offre aucun intérêt sur le plan théorique. Son principal rôle a été de vérifier la démonstration du théorème fondamental de la forêt graphe partiel minimale. Elle s'applique au problème C quand on a une seule zone de recouvrement à partitionner en deux sous-ensembles rattachés chacun à une région postale.

#### PRINCIPE

Prendre la plus petite somme des longueurs de tous les arcs des deux arbres calculés  $2^n$  fois où  $n$  est le nombre des sommets de la zone de recouvrement.

Dans le cas de la figure 21 par exemple les deux arbres minimaux sont construits  $2^6$  fois puisqu'il existe six sommets (X12, X14, X15, X16, X17 et X23) dans la zone de recouvrement. A chaque construction de la forêt on note  $S_1$  et  $S_2$  sommes des longueurs de tous les arcs des deux arbres puis

$$S_k = S_i + S_j \quad \text{avec } i = j = k = 1, \dots, 64 \quad \text{où } i \neq j$$

Les arbres correspondants à :

$$\min_{k \in [1, 64]} (S_k)$$

sont ceux du découpage optimal.

REMARQUE : Les résultats de la programmation de cette méthode

exhaustive confirme ceux obtenus par l'algorithme de la forêt minimale.

#### REMARQUE IMPORTANTE

Le problème C englobe les problèmes A et B et donne une affectation fine et non globale de la "zone de recouvrement" (\*). On entend par là le fait de ramener le choix de l'affectation des sommets au niveau bureau par bureau de poste. Rappelons que le problème 7 du paragraphe 4.C.2 est un cas analogue à celui du problème B puisqu'il s'agit de rattachement globale de sous-ensembles  $z_u, z_v, z_r, z_s$ , etc... (figure 10) de la zone de recouvrement Z. La technique utilisée dans ces deux problèmes n'est toutefois pas la même.

#### 5.4. PROBLEME D : ALIMENTATION FONCTIONNELLE D'UN POLE (\*\*)

##### 5.4.1. Enoncé.

On se donne :

- un nombre fini de centres de tri, de gares et d'aéroport(s) représentés par des sommets dans la figure 25 ;
- une quantité de courrier à chaque sommet ;
- et une (ou des) contrainte horaire à chaque sommet nécessaire à l'acheminement du courrier;

On demande d'établir :

Les lois d'arrivées du courrier à un pôle donné.

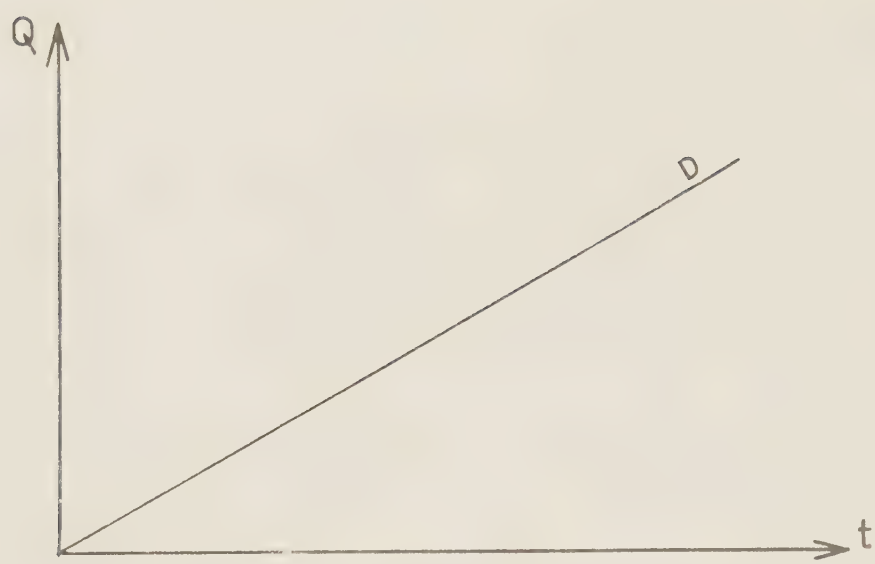
##### 5.4.2. Arborescence minimale de base variable - Alimentation fonctionnelle d'un pôle.

Pour un équipement donné le Centre de Tri a son débit horaire qui peut être représenté par une droite  $D^{(***)}$  figure 23 :

(\*) on se place dans le contexte d'une seule zone de recouvrement engendrée par deux régions postales. Dans le problème 7 §4.C.2 il s'agit en revanche d'un nombre de régions beaucoup plus grand mais fini.

(\*\*) pôle : centre de tri, gares ou aéroport.

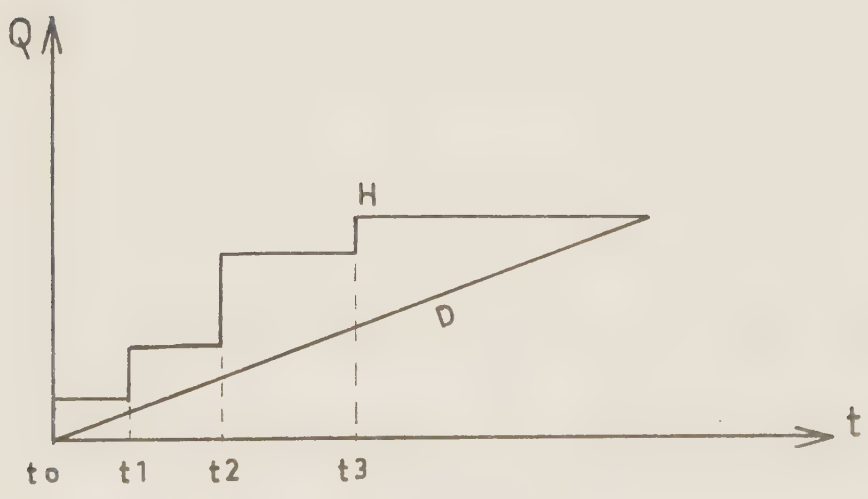
(\*\*\*) la pente, en réalité, n'est constante que dans des intervalles bien définis.



LÉGENDE  
Q = QUANTITÉ DE COURRIER  
t = TEMPS.  
D = DÉBIT HORAIRE.

Figure 23

Si tous les pôles étaient reliés directement (structure radiale) à un pôle donné (pôle 01 par exemple de la figure 25) , le courrier, compte tenu des contraintes, arrive au pôle au rythme suivants (figure 24) :



LÉGENDE  
H = HISTOGRAMME CUMULÉ  
Q = QUANTITÉ GLOBALE  
DU COURRIER  
D = DÉBIT HORAIRE.  
t = TEMPS (UNE FRACTION  
DE LA JOURNÉE PAR  
EXEMPLE)

Figure 24

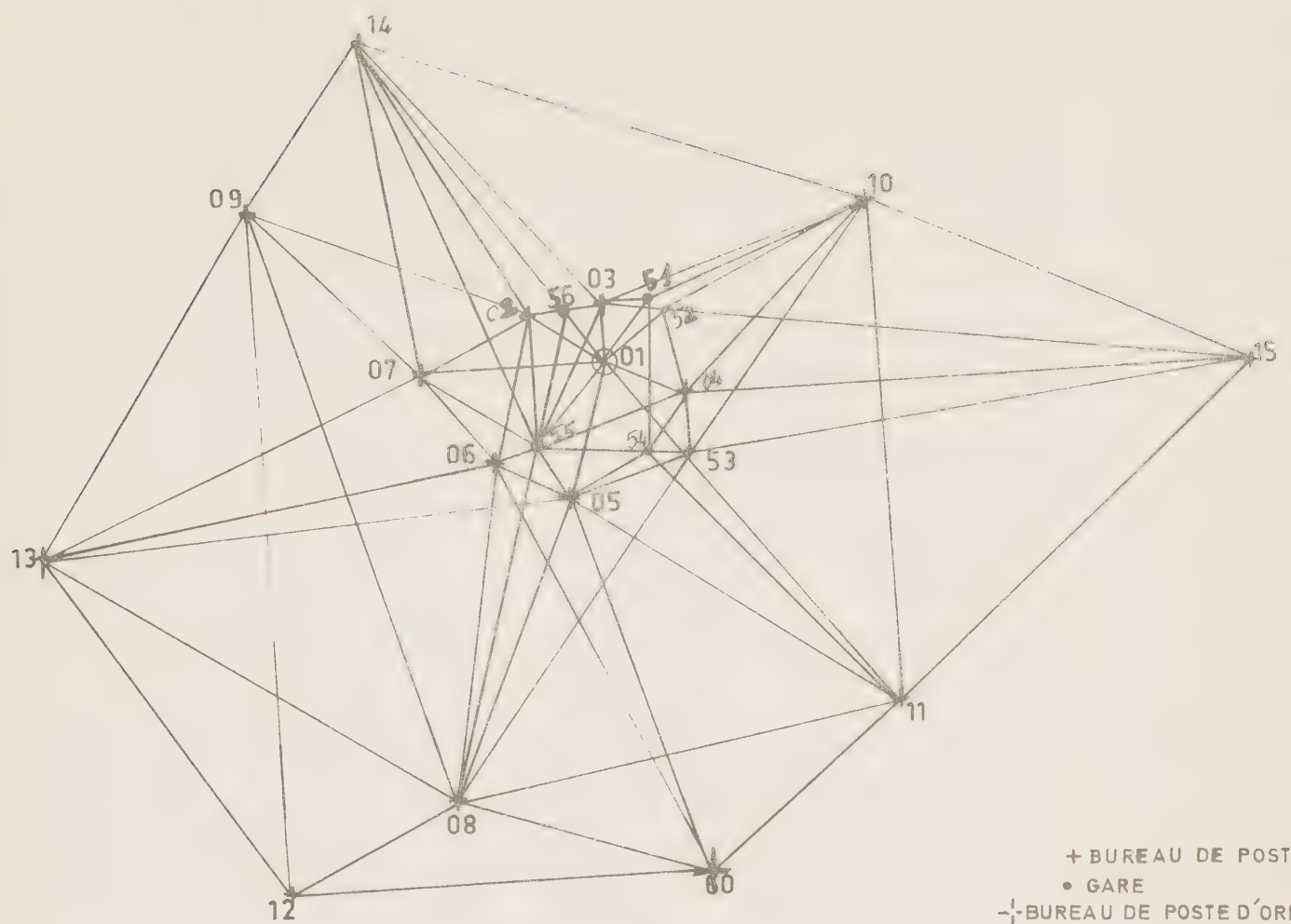


Figure 25 : Réseau où Paris est considéré comme pôle récepteur du courrier des autres pôles.

Le courrier (figure 24) arrive certes suffisamment tôt pour être traité, mais son acheminement est trop coûteux (moyens de transport mis en oeuvre et distance parcourue trop grands). D'où l'idée de construire à partir d'un graphe de  $n$  points des arborescences de base variable à la racine (pôle considéré), c'est-à-dire en rattachant certains des  $n-1$  autres points directement à cette racine : on connecte d'abord, par des arcs,  $C_{n-1}^2$  fois deux sommets parmi les  $n-1$  autres pris deux à deux en construisant à chaque fois l'arborescence autour de la base considérée (\*).

(\*) par la méthode BBB qui s'adapte mieux ici : Kruskal I tel qu'il est programmé ne s'applique pas.

On opère de même en combinant les  $(n-1)$  points 3 à 3 , puis 4 à 4 etc... Soit  $C_{n-1}^3$  ,  $C_{n-1}^4$  , ...,  $C_{n-1}^{n-1}$  le nombre d'arborescences obtenues. Le cas  $C_{n-1}^{n-1}$  représente, on le sait, la structure radiale.

A chaque construction d'arborescence on établit son histogramme cumulé qui ne doit pas intercepter (contrainte fondamentale) la droite D sinon on a une rupture d'alimentation des machines (\*).

#### 5.4.3. Procédure d'application des arborescences.

- Construire toutes les arborescences comme définit précédemment;
- Ordonner ces arborescences par ordre de longueur croissante;
- Prendre la plus courte et tracer son histogramme cumulé correspondant,
- si l'histogramme est en dessous de la droite D , ou s'il la coupe, passer à l'arborescence suivante ;
- Poursuivre la procédure jusqu'à remplir la contrainte fondamentale de se trouver au dessus de la droite D .
- Arrêter alors la procédure, l'arborescence minimale obtenue correspondra à celle de plus courte longueur parmi toutes celles qui respectent la contrainte fondamentale.

#### REMARQUE

Cette procédure donne une arborescence correspondant à l'alimentation optimale d'un pôle; cela correspond à une indication de référence car dans la pratique on préférerait prendre une arborescence de longueur plus grande pour se préserver contre les conséquences de toutes pertes de temps dues à de petites anomalies dans le fonctionnement du système d'exploitation (transport, manutention, panne machine etc...).

L'intérêt de la procédure réside toutefois dans l'évaluation chiffrée de ce qu'on perd par rapport à l'optimum, ce qui permet de mieux

---

(\*) Ce qui risque, quand ce n'est pas prévu dans le planning, de nuire à la qualité de service.

pondérer certains paramètres qui rentrent dans le choix du programme fonctionnel d'exploitation.

## 5.5. PROBLEME E : PROCEDURE POSTALE DE DETERMINATION DE LA "INFLUENCE"

### 5.5.1. ENONCE

On se donne :

- Un centre de tri  $c_j$  caractérisé par :
  - $H_j$  sa plage horaire de traitement du courrier;
  - $D = f(K)$  son débit de traitement qui est fonction du nombre  $K$  de modules (machines) mis en service;
  - $T_j$  son heure de coupure<sup>(\*)</sup>.
- Un nombre fini de bureaux de poste<sup>(\*\*)</sup>, borné entre un nombre minimum  $m$  et un nombre maximum  $M$  de bureaux, et caractérisés chacun par :
  - $Q_i$  = sa quantité de courrier du bureau  $i$  ;
  - $L_{ij}$  = heure de départ au plus tôt du courrier ;
  - $T_{ij}$  = temps de transport entre le bureau  $i$  et le centre  $j$  .

On demande de déduire :

- la liste des bureaux connectables au centre de tri  $c_j$  telle
  - qu'il n'existe pas d'interruption d'alimentation au centre  $c_j$  ;
  - que les contraintes (heure de coupure, plage horaire) soient respectées.
- le nombre correspondant de machines (trieuses et pré-trieuses) mises en service.

(\*) heure à laquelle on arrête le tri d'une catégorie de courrier pour passer à une autre. On se place, dans ce problème, dans le cadre du traitement du courrier collecté et distribué au niveau régional noté G.A.P. Une partie de ce courrier provient, ou est destiné, à l'extérieur du G.A.P.

(\*\*) Préfecture ou sous-préfecture par exemple.

### 5.5.2. Principe de résolution.

On part d'un fichier contenant les données de l'énoncé qui se rapportent à tous les bureaux se trouvant à une distance maximum  $d$  du centre  $c_j$ .

Tout bureau lu dans le fichier et répondant aux critères suivants appartient à la "zone d'influence" du centre  $c_j$  si :

$$(1) \quad L_{ij} + T_{ij} \leq T_j - \frac{Q_i}{D}$$

$$(2) \quad TR + \theta_{ij} \leq H_j$$

$$\text{avec} \quad \left| \begin{array}{l} \theta_{ij} = \frac{Q_i}{D} \\ TR = \sum_i \theta_{ij} \end{array} \right.$$

- (3) La loi d'arrivée du courrier se fait de façon à permettre le tri dès le début de la plage horaire jusqu'à l'heure de coupure sans tolérer de rupture d'alimentation.

### 5.5.3. Remarques.

- Le nombre  $K_{\max}$  de modules étant connu, on commence par  $K=1$  et on incrémente d'une unité jusqu'à  $K_{\max}$  à chaque fois que  $TR$  dépasse  $H_j$ . A chaque incrémentation on recommence à partir du premier bureau.
- La "zone de distribution" est plus facile à déterminer. Il suffit d'éliminer par un test tout bureau se trouvant à une distance telle que le courrier ne puisse lui parvenir avant l'heure de distribution.
- La "zone d'influence" du centre  $c_j$  sera l'intersection des "zones collecte et distribution".

### 5.5.4. EXEMPLE D'APPLICATION (Listing ci-dessous).

On notera la souplesse de la procédure quant à la partie conversationnelle du listing où :

CODE = indique le numéro de chaque bureau  $i$  retenu

HEURE = celle d'arrivée du courrier de  $i$  au centre  $c_j$

QUANTITE CUMULEE (au centre  $c_j$ ) =  $Q_i$  du premier bureau  
s'ajoute à celle du second etc...

HEURE DE FERMETURE DU CENTRE EN HEURES MIN

FRAPPEZ L'HEURE? 24

ET LES MINUTES? 00

PLAGE HORAIRE EN HEURES MINUTES

FRAPPEZ LES HEURES? 6

ET LES MINUTES

? 0

NOMBRE DE BUREAUX RATTACHABLES 22 NOMBRE DE MODULES NECESSAIRES 5

| CODE | HEURE | QUANTITE CUMULEE |
|------|-------|------------------|
| 322  | 19 15 | 89.8             |
| 324  | 19 30 | 121.4            |
| 325  | 20 5  | 139.0            |
| 323  | 20 10 | 158.4            |
| 112  | 20 23 | 198.1            |
| 240  | 20 36 | 212.8            |
| 162  | 20 58 | 224.8            |
| 110  | 21 3  | 267.6            |
| 175  | 21 9  | 284.5            |
| 111  | 21 10 | 320.2            |
| 321  | 21 13 | 331.2            |
| 149  | 21 22 | 347.6            |
| 238  | 21 46 | 395.5            |
| 241  | 21 53 | 426.8            |
| 176  | 21 59 | 458.3            |
| 237  | 22 0  | 494.8            |
| 159  | 22 1  | 518.1            |
| 148  | 22 7  | 529.9            |
| 161  | 22 11 | 558.0            |
| 299  | 22 15 | 597.0            |
| 221  | 22 17 | 609.7            |
| 242  | 22 27 | 628.7            |

AT LINE NØ. 630/ STØP

USED 19.00 UNITS



## CHAPITRE VI

### P R O G R A M M A T I O N

#### 6.1. ALGORITHME DE FORET MINIMALE

##### 6.1.1. Notation

- G.A.P. = correspond à  $C_i$  "zone propre" dans (§ 2.1)  
Z.C. = correspond à  $Z_s$  dans (§ 2.1)  
KG = numéro du G.A.P.  
CV = numéro (code de la ville)  
L = longueur d'arc  
MG = nombre maximum de centres  
NZ = nombre maximum de Z.C.  
D = nombre maximum de sommets rattachables à un centre  
Z = nombre maximum de bureaux dans une Z.C.  
JG = compteur pour les G.A.P.  
JZ = compteur pour les Z.C.  
V = vecteur dans lequel on range les codes villes dans l'ordre  
1, 2, ..., N .  
NG×D = regroupe les vecteurs V relatifs à chaque G.A.P.  
NG×Z = regroupe les vecteurs V relatifs à chaque Z.C.  
NG×Z = cette seconde matrice sert à noter les rattachements choisis  
pour chaque ville de chaque Z.C.  
IDG = désigne la matrice NG×D  
IDZ = désigne la matrice NG×Z  
IRZ = désigne la seconde matrice NG×Z  
L(Z, 2, NZ) = matrice servant à noter dans F les arcs, appartenant aux  
Z.C , entre sommets dont l'affectation à un  $C_i \forall i$  n'est pas  
encore connue.

### 6.1.2. Programme de forêt minimale.

AB140P

```

10C      *****
20C      * FORÊT OPTIMALE (1ERE PARTIE) *
30C      *****
40C
50C
26C
100      $FILE DS114D,DS115D,DS116D
110      REAL FICNOM,FICTEST
120      $OPTINT
130      DIMENSION LC(6,2,4),JL(4),V(6),JG(3),JZ(4),TNG(30,30,3)
140      +IDG(6,30),TNZ(6,6,6),IDZ(4,6),LZ(4,6)
150      COMMON LC,JL,V
160      ZF1:FORMAT(5X,A6)
170      ZF2:FORMAT(/,"*** ERREUR DE LABEL : ",A6," AU LIEU DE ")
180      FGV:FORMAT(5X,I2,X,I2)
190      FGL:FORMAT(5X,2I2,X,I1,X,2I1,2X)
200      FZNF:FORMAT(4X,I2,3X,I2,3X,I2)
210      PRINT,"--- NOMBRE DE G.A.P",; INPUT,NG
220      PRINT,"--- NOMBRE DE ZONES COMMUNES",; INPUT,NZ
230      PRINT,"",; "    GAP SUR1 SUR2",;
240C
250C      CENTROLE LABELS ET INTRODUCTION DONNEES
260C      -----
270      DO Z20,N=1,3
280      READ,FICTEST
290      READ(N,ZF1)FICNOM; IF(FICNOM-FICTEST)Z10,Z20,Z10
300      Z10:PRINT ZF2,FICNOM,FICTEST
310      ZARRET:PRINT,"    ARRET DU TRAITEMENT."; STOP
320      Z20:CONTINUE
330      LEC1:IF(ENDFILE 1) LCT1,LEC2
340      LCT1: READ(1,FGV) KG,CV
350          JG(KG)=JG(KG)+1
360          IDG(KG,JG(KG)) = CV
370          GO TO LEC1
380      LEC2:IF(ENDFILE 2) LCT2,INIT
390      LCT2: READ(2,FGV) KZ,CV
400          JZ(KZ)=JZ(KZ)+1
410          IDZ(KZ,JZ(KZ)) = CV
420          GO TO LEC2
430      INIT:DO INITG,KG=1,NG
440          LIMG=JG(KG)
450          DO INITG,L=1,LIMG
460              INITG: TNG(L,L,KG)=1
470          DO INITZ,KZ=1,NZ
480              LI=LZ=JZ(KZ)
490              DO INITZ,L=1,LIMZ
500              INITZ: TNZ(L,L,KZ)=1

```

001400: COT 1 INH 0

```

5100
5200     ETAPE PRELIMINAIRE
5300     -----
540     LL:READ(C,F,L) V1,V2,C1,C2,N
550     IF(CODE=2) AMM,NN,MN
560     AMM: LINC=JC(C)
570     DO IDMM,I=1,LIMG
580         IF(IDC(M,I)-V1) V2MM,C1MM,V2MM
590         C1MM: C1=1
600         IF(C2) IDMM,IDMM,RL1
610         V2MM: IF(IDC(C,I)-V2) C2MM,C2MM,IDMM
620         C2MM: C2=1
630         IF(C1) IDMM,IDMM,LL1
640         CMM: CMM=IDMM
650     PRINT FCER,V1,V2,M
660     FCER:FORMAT("*** UNE DES VILLES ",I2,X,I2," N'APPARTIENT
670     +PAS AU GAP ",I2)
680     GOTO ZALRET
690     RL1: DO INL1MM,L=1,C1
700         IF(TNG(L,C1,M)) INL1MM,INL1MM,SL1MM
710         C1L1M:CONTINUE
720     SL1MM:L1=L
730     DO INL2MM,L=1,C2
740         IF(TNG(L,C2,M)) INL2MM,INL2MM,SL2MM
750         C2L2M:CONTINUE
760     SL2MM:L2=L
770     L=MAX(L1,L2)
780     C=MINO(L1,L2)
790     L1=L2=0
800     IF(P=0) GO=F,ACDE
810     LINC=JC(C)
820     DO REMM,C=1,LIMG
830         TNG(C,C,M)=TNG(C,C,M)+TNG(P,C,M)
840     REMM: TNG(P,C,M)=0
850     PRINT FCINF,M,V1,V2
860     C1=C2=0; GOTO TF
870     REMM:C1=C2=0; GOTO L1
8800
8900     TRAITEMENT LIATC= N
9000     -----
910     NN:LINC=JC(N)
920     DO IDNN,I=1,LINZ
930         IF(IDZ(N,I)-V1) VPMN,C1NN,V2NN
940         C1NN: C1=1
950         IF(C2) IDNN,IDNN,NV
960         V2NN: IF(IDZ(N,I)-V2) IDNN,C2NN,IDNN
970         C2NN: C2=1
980         IF(C1) IDNN,IDNN,NV
990     IDNN:CONTINUE
1000     PRINT FCERP,V1,V2,N

```

AR140P CONTINUED

```

1010 *CEP2:FEWAL("488 LINE DES VILLES ",C1,N,IP," N° PATIENT
1020 + PASALA ZONE COMMUNE ",IP)
1030 GOTO ZARRET
1040 RV:RV1=IRZ(N,C1)
1050      RV2=IRZ(N,C2)
1060 IF(RV1) TR22,TR22
1070 TR21:IF(RV2) MV11,MV11
1080 C1=C2=0
1090 IF(CV1=RV2) LL,RV12,LL
1100 MV12:=RV13; GOTO APT
1110 MV11:=MV13; GOTO COT
1120 TR22: IF(RV2) ANN,ANN
1130      FAV
1140      GO TO ENN
1150C
1160C      TRAITEMENT LIAISON NN
1170C      -----
1180 GNN:VS=V2; C=C1; GOTO FIXLIN
1190 CNN:VS=V1; C=C2
1200 FIXLIN: LING=JG(M)
1210 DO RECCOL,IC=1,LING
1220      IF(IDG(M,IC)-VS) RECCOL,RECLIG
1230      RECCOL:CONTINUE
1240 REIG:DO RECLIG,L=1,IC
1250      IF(TNG(L,IC,M)) RECLIG,RECLIG,LNN
1260      RECLIG:CONTINUE
1270 LNN:LL=L
1280 DO RECLZ,L=1,C
1290      IF(TNZ(L,C,N)) RECLZ,RECLZ,RAZS
1300      RECLZ:CONTINUE
1310 RAZS:S=0; LINZ=JZ(N)
1320 DO RECCZ,J=1,LINZ
1330      IF(IRZ(L,J,N)) RECCZ,RECCZ
1340      S=S+1
1350      V(S)=IDZ(N,J)
1360      IDG(M,JG(M)+1)=V(S)
1370      IRZ(N,J)=M
1380      JG(M)=JG(M)+1
1390      TNG(LM,JG(M),M)=1
1400      RECCZ:CONTINUE
1410 PRINT F0INF,M,V1,V2
1420 IF(C-1) RAZLC,RAZLC
1430 CALL LS(M,N,S)
1440 RAZLC:L1=L2=0; C1=C2=0; GOTO IF
1450 ANN: DO RECL1,L=1,C1
1460      IF(TNZ(L,C1,N)) RECL1,RECL1,RL1NN
1470      RECL1:CONTINUE
1480      RL1NN:L1=L
1490      DO RECL2,L=1,C2
1500      IF(TNZ(L,C2,N)) RECL2,RECL2,RL2NN

```

## AB140P CONTINUED

```

1510          RECL2:CONTINUE
1520          RL2NN:P=MAXO(L1,L2)
1530          Q=MINO(L1,L2)
1540          C1=C2=0
1550          L1=L2=0
1560          IF(P-Q) LL,LL
1570          LIMZ=JZ(N)
1580          DO RENN,C=1,LIMZ
1590              TNZ(Q,C,N)=TNZ(Q,C,N)+TNZ(P,C,N)
1600          RENN: TNZ(P,C,N)=0
1610          JL(N)=JL(N)+1
1620          LC(JL(N),1,N)=V1
1630          LC(JL(N),2,N)=V2
1640          GO TO LL
1650  SUNE AP141P

```

---

USED        7.00 UNITS.

A61411

```

100 *****
110 * FURET OPTIMALE (2EME PARTIE) *
120 *****
130
140
150 MN: LIMZ=JZ(N) ; LIMG=JG(M)
160 DO COLMN,I=1,LIMZ
170 IF(IDZ(N,I)-V2) COLMN,C2MN
180 COLMN:CONTINUE
190 PRINT F0F03,V2,N
200 FOR R:R=1:RATC("123456789",I2,"") N'APPARTIENT PAS A LA LIGNE CDE
210 WRITE " ",I2
220 GOTO 230
230 C2MN:C2=I
240 IF(IRE(N,C2)-1) TRMN,L2C2MN,TRMN
250 AC2MN:C2=0; GOTO 260
260 TRMN:IF(IRE(N,C2)) ANC2,IDCV1
270 ANC2:C2=0; GOTO 280
280 IDCV1: DO IDCV1,C=1,LIMG
290 IF(IDG(M,C)-V1) IDCV1,IDLV1
300 IDCV1:C:CONTINUE
310 IDLV1: DO IDLV1,L=1,C
320 IF(ING(L,C,I)) IDLV1,IDL1,IDLV2
330 IDLV1:CONTINUE
340 IDLV2: L1=L
350 DO IDV2,L=1,C2
360 IF(117(L,C2,N)) IDV2,RV2,RLS
370 RV2:CONTINUE
380 RLS:S=0
390 DO RATT,J=1,LIMZ
400 IF(TNZ(L,J,N)) RATT,RATT
410 S=S+1
420 V(S)=IDZ(N,J)
430 IFZ(N,J)=M
440 JG(M)=JG(M)+1
450 IDG(M,JG(M))=V(S)
460 TNZ(L,JG(M),I)=1
470 RATT:CONTINUE
480 PRINT F0INF,M,V1,V2
490 IF(S-1) C20,C20
500 IF(LS(C,N,S))
510 C20:C2=0; GOTO 230
520
530 TRAITEMENT LIAISON MN
540 -----
550 TR: DO TR1,I=1,N2

```

AB141P CONTINUED

```

510          L=JZ(1)
520          DO TF1,J=1,L
530              IF(IRZ(1,J)) LL,LL
540          TF1:CONTINUE
550          DO TF2,I=1,NG
560              LIMG=JG(NG)
570              DO TF2,J=1,LIMG
580                  IF(TNG(1,J,I)) LL,LL
590          TF2:CONTINUE
600          PRINT,11,"*** FIN TRAITEMENT UT AB140P."
610          END
6200
6300          CP 140P 1 11
6400          -----
650          SUBROUTINE F012(X,Y,LIM)
660          $OPTINT
670          DIMENSION L (4,2,4), JL(4),V(6)
680          COMMON LC,JL,V
690          F012:FORMAT(4X,I2,3X,I2,3X,I2)
700          LIMJ=JL(Y)
710          DO TF2,J=1,LIMJ
720              DO TF1,K=1,LIM
730                  IF(LC(J,1,Y)-V(K)) T2,EDT,T2
740                  T2:IF(LC(J,2,Y)-V(K)) TF1,EDT
750                  TF1:CONTINUE
760                  EDT:PRINT F012,X,LC(J,1,Y),LC(J,2,Y)
770                  TF2:CONTINUE
780              RETURN
7900
800          $DATA
810          DS114P DS115P DS116P

```

---

USED 4.67 UNITS

6.1.3. Exemple Numérique.Résultats relatifs à deux G.A.P. et une Z.C.

Première colonne : numéro de l'arbre  $A_1$  dans F

Deuxième colonne : première extrémité d'arc (CV)

Troisième colonne : deuxième extrémité d'arc (CV)

Ainsi on lit à la première ligne, par exemple, que l'arbre  $A_1$  de F contient l'arc repéré par ses extrémités 4 et 5 .

AB140P

IN AB141P

--- NOMBRE DE G.A.P? 2

--- NOMBRE DE ZONES COMMUNES? 1

GAP BUR1 BUR2

|   |    |    |
|---|----|----|
| 1 | 4  | 5  |
| 2 | 29 | 30 |
| 1 | 11 | 12 |
| 2 | 31 | 32 |
| 1 | 21 | 22 |
| 2 | 25 | 26 |
| 1 | 13 | 14 |
| 2 | 20 | 21 |
| 1 | 6  | 10 |
| 1 | 5  | 6  |
| 1 | 14 | 23 |
| 1 | 3  | 18 |
| 2 | 27 | 28 |
| 1 | 4  | 15 |
| 2 | 27 | 29 |
| 2 | 32 | 34 |
| 1 | 7  | 8  |
| 2 | 19 | 28 |
| 2 | 19 | 20 |
| 1 | 18 | 13 |
| 1 | 12 | 23 |
| 1 | 9  | 10 |
| 2 | 29 | 31 |
| 2 | 19 | 16 |
| 2 | 22 | 25 |
| 1 | 18 | 15 |
| 2 | 25 | 24 |
| 2 | 34 | 17 |
| 2 | 31 | 33 |
| 1 | 1  | 3  |
| 1 | 8  | 9  |
| 1 | 1  | 2  |

## 6.2. PROCEDURE EXHAUSTIVE DE VERIFICATION DES RESULTATS DE LA FORET MINIMALE. (§6.1)

### 6.2.1. Principe :

Partons d'un graphe connexe dont on veut extraire une forêt minimale de  $p=2$  composantes (§ 6.2.3.) . Soit  $N$  le nombre de sommets de  $Z$ ,

- On détermine la liste de toutes les répartitions possibles ( $2^N$  répartitions) ;
- On note la longueur de forêt minimale correspondante à chaque répartition ;
- La longueur la plus courte de ces  $2^N$  répartitions (ou décompositions) correspond à celle de la forêt minimale recherchée.

### 6.2.2. Remarques.

Cette procédure exhaustive présente un double intérêt :

1°/ elle vérifie les résultats obtenus par l'algorithme de forêt minimale (§ 4.A.4.1) ;

2°/ elle mesure l'écart de longueur de  $F$  entre :

- la longueur de  $F$  obtenue par la décomposition optimale ;
- et la longueur de  $F$  obtenue par l'une quelconque des  $(2^N-1)$  autres décompositions susceptibles d'être imposées.

En revanche elle demande un temps machine beaucoup plus grand pour obtenir les mêmes résultats que l'algorithme de forêt minimale ,  
(445,33 unités au lieu de 43,67 unités).



### 6.2.3. Programme de la procédure exhaustive.

AB130P

```

10C *****
20C * PROCEDURE EXHAUSTIVE DE VERIFICATION DES RESULTATS*
25C * OBTENUS PAR L'ALGORITHME DE FOREL MINIMALE *
30C * ( PARTIE 1 ) *
40C *****

200 $FILE AB103D,AB104D,AB105D,AB106D,AB107D,AB110D,AB111D,AB112D
100 INTEGER A(50,50) ; COMMON A
110 INTEGER INT(10) ; COMMON INT
120 INTEGER V(50) ; COMMON V
130 INTEGER NP,N,FIC1,FIC2,FIC3,FIC4,NL,NC0
140 COMMON NP,N,FIC1,FIC2,FIC3,FIC4,NL,NC0
150 INTEGER SOM1,SOM2,TSCM
160 INTEGER CF1(10),TCF1(10)
170 INTEGER CF2(10),TCF2(10)
180 INTEGER ID(4)
190 INTEGER MEM(10),TAMP(50)
200 FIC1=1 ; FIC2=2 ; FIC3=3 ; FIC4=6 ; NL=0
210 READ(1,F0ID) (ID(I),I=1,4)
220 READ(1,F0ID) (ID(I),I=1,4)
230 F0ID:FORMAT(3X,4A2)
240 PRINT," CONFIGURATIONS ET REPARTITIONS OPTIMALES
250 PRINT," POUR LYON ET CLERMONT"
260 PRINT,
270 CALL RENSMO
280 NC0=N-NP
290 DO MEMORD,J=1,NC0
300 MEMORD: MEM(J)=V(NP+J)
310 CALL CALC00(NP,N,0)
320 TF81: IF(ENDFILE 8) LCINT,FING1
330 LCINT:READ(8,F0INT) (INT(I),I=1,NC0)
340 F0INT:FORMAT(3X,2X,10I1)
350 CALL CALC00(NP,N,0)
360 GO TO 100
370 FING1: ENDFILE 8
380 REWIND 8
390 REWIND 6
400 FIC1=2 ; FIC2=3 ; FIC3=7 ; NL=0
410 READ(2,F0ID) (ID(I),I=1,4)
420 CALL RENSMO
430 DO BCLTR,J=1,NC0
440 DO REM,I=1,50
450 REM:TAMP(I)=A(NP+J,I)
460 DO REC,K=J,NC0
470 IF(V(NP+K)-MEM(J)) REC,TRST,REC
480 REC:CONTINUE
490 TRST: TAMPV=V(NP+J) ; V(NP+J)=V(NP+K) ; V(NP+K)=TAMPV
500 DO TRF,I=1,50

```

AB1310: AB1311D

```

510          A(NP+J,I)=A(I,NP+J)=A(NP+K,I)
520          A(NP+K,I)=A(I,NP+K)=TAMP(I)
530  TRF:CONTINUE
540  BCLTR:CONTINUE
550  DO INTE=I-1,PCE
560      INTE=INT(I)=1
570  CALL CALCOO(NP,N,0)
580  TF82: IF(ENDFILE 8) LCINT2,FING2
590  LCINT2: READ(8,FPOINT) (INT(I),I=1,NCI)
600  DO OPP,I=1,NCI
610      OPP:INT(I)=1-INT(I)
620  CALL CALCOO(OPP,N,0)
630  IF(T) T=0
640  CLCTR: ENDFILE 7
650  REWIND 1 ; REWIND 4 ; REWIND 7
660  READ(6,F=1) SOM1,(CF1(I),I=1,10)
670  READ(7,F=1) SOM2,(CF2(I),I=1,10)
680  FOMIN:FORMAT(3X,X,I3,10I1)
690  TSOM = SOM1+SOM2
700  DO MEMIN,I=1,10
710      TCF1(I)=CF1(I)
720      MEMIN:TCF2(I)=CF2(I)
730  TF6: IF(ENDFILE 6) RECMIN,SORMIN
740  RECMIN: READ(6,FEMIN) SOM1,(CF1(I),I=1,10)
750      READ(7,FOMIN) SOM2,(CF2(I),I=1,10)
760  SOM=SOM1+SOM2
770  IF(TSOM-SOM) TF6,TF6,MODSOM
780  MODSOM:TSOM=SOM
790  DO MODMIN,I=1,10
800      TCF1(I)=CF1(I)
810      MODMIN:TCF2(I)=CF2(I)
820  IF(T) TF6
830  IF(1) MN2,I=1,NCI
840      MN2:INT(I)=TCF2(I)
850  PRINT,"          GAP LY="
860  PRINT,"          POINTS  DISTANCE."
870  CALL CALCOO(NP,N,1)
880  FIC1=1 ; FIC2 = 4
890  READ(1,F=1) (ID(I),I=1,4)
900  CALL RENSME
910  DO MN1,I=1,NCI
920      MN1: INT(I)= TCF1(I)
930  PRINT,↑
940  PRINT,"          GAP CLEARNOT"
950  PRINT,"          POINTS  DISTANCE"
960  CALL CALCOO(NP,N,1)
970  GUSE AB131F
1010  STOP
1020  SUBROUTINE RENSME
1030  INTEGER A(50,50) ;COMMON A

```

## AB130P CONTINUED

```

1040  INTEGER INT(10) ; COMMON INT
1050  INTEGER V(50) ; COMMON V
1060  INTEGER NP, P, FIC1, FIC2, FIC3
1070  COMMON NP, N, FIC1, FIC2, FIC3
1080  INTEGER FIC4, NL, NC0 ; COMMON FIC4, NL, NC0
1090  INTEGER P1, P2, D, P, AIG, M1, P2, DC
1100  DO BCLOA, I=1, 50
1110  DO BCLOA, J=1, 50
1120      BCLOA: A(I, J)=0
1130  DO BCLDV, I=1, 50
1140      BCLDV: V(I)=0
1150      NP=0
1160  LECTR1: READ(FIC1, FICINT) P1, P2, D
1170  FICINT: FORMAT(3X, 3I2, 2X)
1180      P=P1 ; AIG=1
1190  DRFCC: DO RFCC, I=1, 50
1200      IF(V(I)-P) LIE, SUPP, LIE
1210      LIE: IF(V(I)) SUPP, LIE, RECPAT
1220  RECPAT: CONTINUE
1230      PRINT, "ERREUR 1"
1240      SUPP: V(I)=P ; NP= I
1250  SUPP: IF(AIG-1) PRINTP, PRINT2, INSP
1260  PRINT2: M1=I ; AIG=2 ; P=P2
1270      GO TO DPRC
1280  INSP: M2=I ; A(M1, M2)=A(M2, M1)=D
1290      IF(ENDFILE FIC1) LECTR1
1300      DC=NP+1 ; NC=0
1310  LECTR2: READ(FIC2, FOLIA) P1, P2, D, DS
1320  FOLIA: FORMAT(3X, 4I2)
1330      DO RECLI, I=1, 50
1340          IF(V(I)-P1) VOIR2, PSP2, VOIR2
1350          VOIR2: IF(V(I)-P2) RECLI, PSP1, RECLI
1360  RECLI: CONTINUE
1370      PRINT, "ERREUR 2"
1380  PSP2: PS=P2 ; GO TO MEMIK
1390  PSP1: PS = P1
1400  MEMIK: K = I
1410      DO INSC1, I=DC, 50
1420          IF(V(I)-PS) LIEN, SUPC1, LIEN
1430  LIEN: IF(V(I)) INSC1, SUPC1, INSC1
1440  INSC1: CONTINUE
1450  SUPC1: V(I)=PS ; NC=NC+1
1460  DEFIL: I=I
1470      A(K, I)=A(I, K)=D
1480      IF(ENDFILE FIC2) LECTR2
1490  LECTR3: READ(FIC3, FICINT) P1, P2, D
1500      P=P1 ; AIG=1
1510  DRFCC: DO RFCC, I=DC, 50
1520      IF(V(I)-1) LIEN, SUPC2, LIEN
1530  LIEN: IF(V(I)) RFCC, SUPC2, RFCC

```

AB130P CONTINUED

```
1540      DCC:CONTINUE
1550      PRINT,"ERROR 3"
1560      SUBC2: V(1)=P ; NC =NC+1
1570      SPLITC:IF(AIG-1) INSC,PTAC,INSC
1580      PTAC: M1=1 ; AIG=2 ; P=P2
1590      GO TO DCECC
1600      INSC: M2= 1
1610      A(M1,M2)=A(M2,M1)=D
1620      IF(C=FILE PIC3) LECTR3
1630      REWIND FILE3
1640      N = NP+NC
1650      RETURN
1660      END
1670      STOP
```

---

USED      7.17   UNITS

AB131P

```

10C  *****
20C  * PROCEDURE EXHAUSTIVE DE VERIFICATION DES RESULTATS*
25C  * OBTENUS PAR L'ALGORITHME DE FORET MINIMALE          *
30C  *              ( PARTIE 2 )                          *
40C  *****

90C
1670 SUBROUTINE CALCOO(VILP,TOTVIL,IMPR)
1680     INTEGER A(50,50) ; COMMON A
1690     INTEGER INT(10) ; COMMON INT
1700     INTEGER V(50) ; COMMON V
1710     INTEGER NP,N,FIC1,FIC2,FIC3
1720     COMMON NP,N,FIC1,FIC2,FIC3
1730     INTEGER FIC4 ; COMMON FIC4
1740     INTEGER NL,NCO ; COMMON NL,NCO
1750     INTEGER VILP,TOTVIL,NBLI,M,VC
1760     INTEGER T(50),P,Q
1770     INTEGER U(50) NL,TOT,DC
1780     INTEGER LIM,IMP
1790     INTEGER RETR
1800     RETR=0
1810     DO CALRETR,I=1,10
1820         CALRETR:RETR =RETR+INT(I)
1830     M=0 ; TOT=0
1840     DC =VILP+1 ; LIM=TOTVIL-2-RETR
1850     J=1
1860     DO INTER,I=DC,TOTVIL
1870         U(I) = INT(J)
1880         INTER: J=J+1
1890     P=1
1900         DO PREMLG, J=1,TOTVIL
1910         IF(A(1,J)) TINT,PREMLG,TINT
1920         TINT:IF(U(J)) TM1,TM1,PREMLG
1930         TM1:IF(M) MOD1,MOD1,TAIJ1
1940         TAIJ1: IF(A(1,J)-M) MOD1,PREMLG,PREMLG
1950         MOD1: M=A(1,J) ; Q=J
1960         PREMLG:CONTINUE
1970         TOT=TOT+M
1980         T(P)=T(Q)=1
1990         IF(IMPR) SVT,SVT,PRI
2000         PRI: IF(Q-VILP) PROP,PROP,COM
2010         COM:IMP=2
2020         GO TO IMPR1
2030         PROP:IMP=1
2040         IMPR1: CALL IMPRES(IMP,P,Q,NP)
2050         SVT:DO ECLI,NBLI=1,LIF
2060             M=0
2070             DO ECLI,I=1,TOTVIL

```

# AB131P CONTINUED

```

2090 IF(T(I)) BCL1,BCL1,INT3
2095 INT3: IF(U(I)) D=CLJ,B=CLJ,BCL1
2100 DBCLJ:D0=CLJ,J=1,TOTVIL
2110 IF(T(J)) INT3,INT3,BCLJ
2120 INT3: IF(U(J)) TM2,TM2,BCLJ
2130 T32:IF(I) TAIJ3,TAIJ3,TAIJ2
2140 TAIJ2:IF(A(I,J)-M) TAIJ3,CLJ,BCLJ
2150 TAIJ3:IF(A(I,J)) BCLJ,BCLJ,END2
2160 MEMC:V=A(I,J) ; P=1 ; Q=1
2170 BCLJ:CONTINUE
2180 BCL1:CONTINUE
2190 TOT = TOT+1
2200 T(I)=T(I)+1
2210 IF(IMPR) BCL,BCL,PR13
2220 PR13:IF(1-VILP) TOT,TOT,TOT
2230 TOT:IF(1-VILP) CO,CO,CO
2240 CO:IMP=2
2250 GO TO PR13
2260 CO:IMP=1
2270 GO TO PR13
2280 TOT: IF(1-VILP) CP,CP,CP
2290 CP:IMP=2
2300 GO TO PR13
2310 CPQ:IMP=3
2320 PR13: CALL IMPRES(IMP,P,Q,NP)
2330 BCL:CONTINUE
2340 IF(IMPR) MEMCC,MEMCC,RET0UR
2350 MEMCC: NC0=TOTVIL-VILP
2360 NL=NL+1
2370 WRITE(FIC4,F0MEM) NL,TOT,(INT(I),I=1,10)
2380 RET0UR: D0 BCL0T,I=1,50
2390 U(I)=0
2400 BCL0T:T(I)=0
2410 F0MEM:FORMAT(I3,X,I3,10I1)
2420 RETURN
2430 END
2440 SUBROUTINE IMPRES(CAS,X,Y)
2450 INTEGER A(50,50) ; COMMON A
2460 INTEGER INT(10) ; COMMON X INT
2470 INTEGER V(50) ; COMMON V
2480 INTEGER CAS,X,Y
2490 IF(CAS=2) INTE,RATT,EXT
2500 INTE:PRINT F0INT,V(X),V(Y),A(X,Y)
2510 F0INT:FORMAT(19HLIAISON INTERNE *,I2,1H*,I2,6H** ,I2,
2520 +6H **)
2530 GO TO SORTIE
2540 RATT: PRINT FOR,V(X),V(Y),A(X,Y)
2550 FOR:FORMAT(19HLIAISON DE RATT. *,I2,1H*,I2,6H** ,I2,
2560 + 6H **)
2570 GO TO SORTIE

```

AB131P CONTINUED

```
2580 EXT:PRINT F0X,V(X),V(Y),A(X,Y)
2590 F0X:F0RMAT(19HLIAISON EXTERNE  *,I2,1H*,I2,6H**      ,I2,
2600 +          6H      **)
2610 SORTIE: RETURN ; END
```

---

USED 5.83 UNITS

6.2.4. Exemple numérique.

Résultats relatifs aux mêmes G.A.P. (même  $C_i$ ) et à la même Z.C. (même  $Z_s$  de Z) que dans (§ 6.1.3).

Ces résultats sont absolument identiques (cf figure 2, § 5.2.3).

AB130P

IN AB126P  
IN .FIRST

CONFIGURATIONS ET REPARTITIONS OPTIMALES  
POUR LYON ET CLERMONT

GAP LYON

|                  | P0INTS   | DISTANCES |    |
|------------------|----------|-----------|----|
| LIAIS0N INTERNE  | *19*28** | 25        | ** |
| LIAIS0N INTERNE  | *28*27** | 23        | ** |
| LIAIS0N INTERNE  | *27*29** | 24        | ** |
| LIAIS0N INTERNE  | *29*30** | 10        | ** |
| LIAIS0N INTERNE  | *19*20** | 26        | ** |
| LIAIS0N INTERNE  | *20*21** | 20        | ** |
| LIAIS0N INTERNE  | *21*22** | 18        | ** |
| LIAIS0N INTERNE  | *29*31** | 29        | ** |
| LIAIS0N INTERNE  | *31*32** | 18        | ** |
| LIAIS0N INTERNE  | *32*34** | 24        | ** |
| LIAIS0N DE RATT. | *19*16** | 31        | ** |
| LIAIS0N INTERNE  | *22*25** | 31        | ** |
| LIAIS0N INTERNE  | *25*26** | 19        | ** |
| LIAIS0N DE RATT. | *34*17** | 33        | ** |
| LIAIS0N INTERNE  | *25*24** | 33        | ** |
| LIAIS0N INTERNE  | *31*33** | 34        | ** |

GAP CLERMONT

|                  | P0INTS   | DISTANCES |    |
|------------------|----------|-----------|----|
| LIAIS0N INTERNE  | * 1* 3** | 35        | ** |
| LIAIS0N INTERNE  | * 3*18** | 23        | ** |
| LIAIS0N INTERNE  | *18* 4** | 24        | ** |
| LIAIS0N INTERNE  | * 4* 5** | 10        | ** |
| LIAIS0N INTERNE  | * 5* 6** | 21        | ** |
| LIAIS0N INTERNE  | * 6*10** | 21        | ** |
| LIAIS0N INTERNE  | *18*13** | 26        | ** |
| LIAIS0N DE RATT. | *13*14** | 19        | ** |
| LIAIS0N EXTERNE  | *14*23** | 23        | ** |
| LIAIS0N INTERNE  | *10* 9** | 27        | ** |
| LIAIS0N EXTERNE  | *23*12** | 27        | ** |
| LIAIS0N DE RATT. | *12*11** | 16        | ** |
| LIAIS0N DE RATT. | *18*15** | 33        | ** |
| LIAIS0N INTERNE  | * 9* 8** | 38        | ** |
| LIAIS0N INTERNE  | * 8* 7** | 24        | ** |
| LIAIS0N INTERNE  | * 1* 2** | 44        | ** |

# REFERENCES

- [ 1 ] C.F.PICARD - Graphes et Questionnaires, Gauthier-Villars 1972.
- [ 2 ] J.B.KRUSKAL - On the shortest spanning subtree of a graph and the traveling salesman problem Proceedings of the American Mathematical Society 71 , 1956.
- [ 3 ] O.BORUVKA - On a minimal problem , Práce Moravské Pridovedecké společnosti, vol. 3, 1926.
- [ 4 ] P.ROSENSTIEHL - L'arbre minimum d'un graphe, in théorie des Graphes, Journées internationales (Rome 1966) p.357-368 , Dunod, Paris 1967.
- [ 5 ] G.CHOQUET - Etude de certains réseaux de routes, C.R. Ac. Sc. Paris, 206, 1938, P.310-311.
- [ 6 ] R.C.PRIM - Shortest connection networks and some generalizations, The Bell Syst. Techno. Journ., 36, n°6, 1957 p.1389-1401.
- [ 7 ] J.ABADIE - Une méthode arborescente pour les programmes non-linéaires partiellement discrets, R.I.R.O. V-3 , 1969 , p.25-49.
- [ 8 ] C.F.PICARD et A.BAKHTAOUI - Algorithmes de forêts minimales, C.R.A.S. Paris (1972).
- [ 9 ] B.ROY - Algèbre moderne et théorie des graphes, Dunod, Paris 1970.
- [ 10 ] C.BERGE - Graphes et Hypergraphes, Dunod, Paris 1970.
- [ 11 ] A.BAKHTAOUI - Variation de la fonction économique du sous-réseau, NSI (Note de Service Intérieure) - 20 237, Décembre 1971 à SOMEPOST, ORLY - 94.  
- Optimalité dans le découpage du territoire, NSI - 20 280 , janvier 1972, SOMEPOST, ORLY - 94.
- [ 12 ] J.C.DERNIANE ET C.PAIR - Problèmes de cheminement dans les graphes, Chap. 7, Monographies d'Informatique, AFCET, Dunod Paris 1971.
- [ 13 ] L.J.WHITE - Minimum covers of fixed cardinality in weighted graphs, S.I.A.M. Journal on Appl. Math., 21, 1971, p.104-113.



UNIVERSITE de PARIS VI

11, Quai Saint Bernard - Paris 5

SERVICE DE LA SCOLARITE

Bureau des Enseignements du 3e Cycle

tél. 336 25.25 et 325.12.21

## PERMIS D'IMPRIMER

DOCTORAT : DOCTORAT D'UNIVERSITE

THESE DE :

NOM : Monsieur BAKHTAOUI

Prénoms : Abdelmajid

né le 19 novembre 1937 à EL AOUN. Dép<sup>t</sup> : Maroc

1ère THESE : "ALGORITHMES DE FORETS MINIMALES, RESEAUX DE TRANSPORT  
ET OPTIMALITE."

2e THESE (Propositions faites par l'Université) :

PRESIDENT : M. VILLE

EXAMINATEURS : M. PICARD

M. PALLU DE LA BARRIERE

Vu et Approuvé

PARIS, le 6 juillet 1972

Le Président

de l'Université de PARIS VI











